

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-292866

(43)Date of publication of application : 05.11.1996

(51)Int.Cl.

G06F 3/14  
H04N 5/00  
H04N 5/445

(21)Application number : 08-073020

(71)Applicant : AT &amp; T IPM CORP

(22)Date of filing : 28.03.1996

(72)Inventor : EICK STEPHEN G  
MATAGA PETER ANDREW  
WALPOLE REBECCA ANNE

(30)Priority

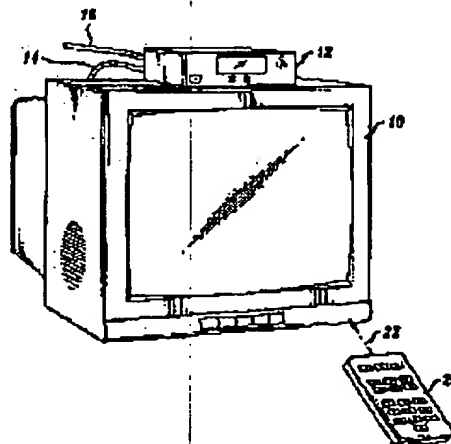
Priority number : 95 412393 Priority date : 28.03.1995 Priority country : US

(54) METHOD FOR DETECTING AND SELECTING DESIRED DATA FROM LARGE AMOUNT OF SCHEDULE OF DATA ITEM BY USING CONTROLLER OF TELEVISION AND REMOTE CONTROLLER OR THE LIKE FOR TELEVISION AND DEVICE THEREFOR

(57)Abstract:

PROBLEM TO BE SOLVED: To allow a viewer to be given a chance to execute selection from the large amounts of item schedules to his satisfaction.

SOLUTION: This device is provided with a filter means including a sub-group designator in a displaying means 10, which selects the sub-group designator by a pointing means for filtering the group, and generates a sub-group by the selected sub-group designator, and the displaying means 10 which displays the expression of a group item belonging to at least one part of the sub-group in the displaying means. The group item is selected by selecting the expression in the displaying means 10 according to the pointing means.



## LEGAL STATUS

[Date of request for examination] 17.12.1999

[Date of sending the examiner's decision of rejection] 03.03.2004

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

BEST AVAILABLE COPY

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-292866

(43) 公開日 平成8年(1996)11月5日

(51) Int.Cl. <sup>8</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 3/14	3 4 0		G 0 6 F 3/14	3 4 0 A 3 4 0 B
H 0 4 N 5/00 5/445			H 0 4 N 5/00 5/445	A Z

審査請求 未請求 請求項の数21 O L (全 97 頁)

(21) 出願番号 特願平8-73020  
(22) 出願日 平成8年(1996)3月28日  
(31) 優先権主張番号 08/412393  
(32) 優先日 1995年3月28日  
(33) 優先権主張国 米国 (U S)

(71) 出願人 595119464  
エイ・ティ・アンド・ティ・アイビーエム・コーポレーション  
アメリカ合衆国, 33134 フロリダ, コーラル ゲーブルズ, ボンス ド レオン  
ブウルヴァード 2333  
(72) 発明者 スチーヴン グレゴリー イック  
アメリカ合衆国 60565 イリノイズ, ネイバーヴィル, ダーネス コート 1413  
(74) 代理人 弁理士 岡部 正夫 (外10名)

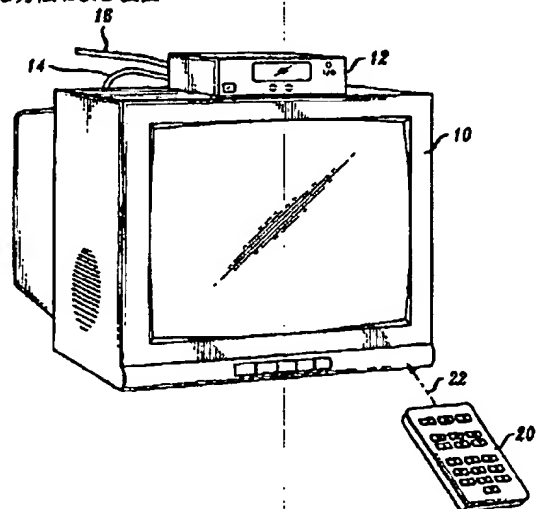
最終頁に続く

(54) 【発明の名称】 テレビおよびテレビ用遠隔制御装置などのコントローラを用いてデータ項目の多量のスケジュールから所望のデータを検出および選択する方法および装置

(57) 【要約】 (修正有)

【課題】 尤大な項目のスケジュールから思い通り選択が行える機会を視聴者に与える。

【解決手段】 表示手段内のサブグループ指定子を含み、グループをフィルタリングするためのポインティング手段によってサブグループ指定子を選択し、選択されたサブグループ指定子によってサブグループを生成するフィルタ手段、及び表示手段内のサブグループの少くとも一部分に属するグループ項目の表現を表示する表示手段を備え、ポインティング手段に応じて表示手段内の表現を選択することにより、グループ項目を選択する。



## 【特許請求の範囲】

【請求項1】 表示手段と、表示手段における位置を特定するための対話型可動式ポインティング手段とを有し、特定された位置の選択を行うシステム内のグループから項目を選択する装置において、

表示手段内のサブグループ指定子を含み、グループをフィルタリングするためのポインティング手段によりサブグループ指定子の選択に反応して、選択されたサブグループ指定子により指定されたサブグループを生成するフィルタ手段と、

表示手段内のサブグループの少なくとも一部分に属するグループ項目の表現を表示する手段と、

ポインティング手段に応じて表示内の表現を選択することによりグループ項目を選択するためのグループ項目選択手段とからなることを特徴とする装置。

【請求項2】 ポインティング手段は、ある表現からそれと隣接する表現に移動可能であることのみを必要とすることを特徴とする請求項1記載の装置。

【請求項3】 表現を表示するための手段は、

単一次元の表現を表示する第1の手段と、

2次元の表現を表示する第2の手段とからなることを特徴とする請求項1記載の装置。

【請求項4】 全体のサブグループの縮小された表現と、表示手段によって現在表示されているグループの一部の縮小された表現におけるインディケーションを表示する手段をさらに有することを特徴とする請求項1記載の装置。

【請求項5】 前記縮小された表現は2次元であることを特徴とする請求項4記載の装置。

【請求項6】 前記対話型可動式ポインティング手段は、

第1の方向へ表示内の位置における変更をコントロールするための第1の対のボタンと、第2の方向へ表示内の位置における変更をコントロールするための第2の対のボタンとからなる遠隔制御装置を含むことを特徴とする請求項5記載の装置。

【請求項7】 前記縮小された表現は3次元表現の2次元表現であり、3次元は少なくともひとつの共通の属性を有する項目の論理スタック内のロケーションであることを特徴とする請求項4記載の装置。

【請求項8】 論理スタックの各項目は、ひとつの共通の属性として視野タイムスロットであることを特徴とする請求項7記載の装置。

【請求項9】 前記対話型可動式ポインティング手段は、

第1の方向へ表示内の位置における変更をコントロールするための第1の対のボタンと、第2の方向へ表示内の位置における変更をコントロールするための第2の対のボタンと、論理スタック内の位置における変化をコントロールするための第3の対のボタンとからなる遠隔制御

装置を含むことを特徴とする請求項8記載の装置。

【請求項10】 番組の情報をセット・トップ・ボックスに導く同様の情報コンダクタを介してセット・トップ・ボックスにより番組スケジュールデータを受信する工程と、

前記セット・トップ・ボックス内のRAM内の前記番組スケジュールデータをフィルタリングする工程と、

前記セット・トップ・ボックスが視聴者に前記番組スケジュールデータの複数の選択を提供する、これに接続されたTV上の第1の対話型表示を示す工程と、

前記視聴者による対話型選択に応じて、番組スケジュールデータの第1のサブグループへ前記番組スケジュールデータをフィルタリングする工程と、

また、前記視聴者による対話型選択に応じて、前記セット・トップ・ボックスが前記番組スケジュールデータをフィルタリングするための第2の複数の選択を有する前記TV上の第2の対話型表示を示す工程と、

前記視聴者による対話型選択に応じて、第2のサブグループへ前記第1のサブグループをフィルタリングする工程と、

また、前記視聴者による第2対話型選択に応じて、前記セット・トップ・ボックスが前記第2のサブグループの各番組項目の表現を有する前記TV上の第3の対話型表示を示す工程とからなることを特徴とする方法。

【請求項11】 前記番組スケジュールデータ受信工程は、

前記セット・トップ・ボックスを介して前記番組スケジュールデータの第1部分を受信する工程と、

後の接触しない時間における前記番組スケジュールデータの第2部分を受信する工程とを更に含むことを特徴とする請求項10の方法。

【請求項12】 前記第2のサブグループの番組項目の表現の対話型強調に応じて、そのタイトルを表示する工程を更に含むことを特徴とする請求項10記載の方法。

【請求項13】 遠隔制御装置の選択ボタンの作動に応じて、前記強調された番組のプレビューを表示する工程を更に含むことを特徴とする請求項12記載の方法。

【請求項14】 前記遠隔制御装置の前記選択ボタンの第2の作動に応じて、前記強調された表現に対応するTV番組を表示するように前記セット・トップ・ボックスを切り替える工程を更に含むことを特徴とする請求項13記載の方法。

【請求項15】 前記遠隔制御装置の前記選択ボタンの第2の作動に応じて、前記強調された表現に対応するTV番組が始まる時にそのTV番組を表示するように前記セット・トップ・ボックスを切り替えるコマンドを記憶する工程を更に含むことを特徴とする請求項13記載の方法。

【請求項16】 少なくとも1週間の期間のための少なくとも300の個々のチャンネルの番組スケジュールデ

ータを受信する工程と、

データベースフォーマットに迅速な分類と検索のためのローカルメモリ内に番組スケジュールデータを記憶する工程と、

番組スケジュールデータのサブグループへユーザの対話式入力に応じて番組スケジュールデータをフィルタリングする工程と、

ユーザーの確認のために番組スケジュールデータのサブグループを表示する工程と、

TVのスクリーン上に映すために番組スケジュールデータのサブグループから番組を対話式に選択する工程とからなることを特徴とする方法。

【請求項17】 少なくとも1週間の期間のための少なくとも300の個々のチャンネルの番組スケジュールデータを受信する工程と、

データベースフォーマットに迅速な分類と検索のためのローカルメモリ内に番組スケジュールデータを記憶する工程と、

番組スケジュールデータのサブグループへユーザの対話式入力に応じて番組スケジュールデータをフィルタリングする工程と、

ユーザーの確認のために番組スケジュールデータのサブグループを表示する工程と、

ビデオレコーディング装置による記録のための番組スケジュールデータのサブグループから番組を対話式に選択する工程とからなることを特徴とする方法。

【請求項18】 そのデータがローカルメモリ内に記憶されている番組の多量のスケジュールから所望の番組を選択する方法において、

それぞれ特定の時間およびチャンネルの番組を表現するカードからなる縦続接続のグループを表示する工程と、前記カードのグループのサブグループの周りに配された選択ウインドウを表示する工程と、

前記カードの縦続接続のグループにより表現された前記番組のサブグループが更に詳しく示される前記カードの縦続接続のグループに隣接する2次元グリッドを表示する工程と、

前記番組のサブグループのひとつを強調する前記選択ウインドウ内の第1の作動エリアを表示する工程と、

前記2次元グリッド内の第2の作動エリアを表示する工程と、前記第2の作動エリアは前記第1の作動エリアにおいて強調された番組の詳細の周りに配され、且つ強調するものであって、

遠隔制御装置からの視聴者の入力へ縦方向の矢印に応じた縦方向へ前記作動エリアを移動する工程と、

前記セット・トップ・ボックスが前記選択を行うまで前記所望の番組へ前記作動エリアを移動させ選択ボタンを作動させることにより所望の番組を選択する工程とからなることを特徴とする方法。

【請求項19】 前記視聴者からの入力により前記選択

ウインドウ外の位置へ前記作動エリアが移動されたあと、前記選択ウインドウを前記作動エリアが移動された接続するサブグループへ移動する工程を更に含むことを特徴とする請求項18記載の方法。

【請求項20】 表示手段と、表示手段における位置を特定するための対話型可動式ポインティング手段とを有し、特定された位置の選択を行うシステム内のグループから項目を選択する装置において、

表示手段内のサブグループ指定子を含み、グループをフィルタリングするためのポインティング手段によりサブグループ指定子の選択に反応して、選択されたサブグループ指定子により指定されたサブグループを生成するフィルタ手段と、

表示手段内のサブグループに属するグループ項目の表現を表示する手段と、

ポインティング手段に応じて表示内の表現を選択することによりグループ項目を選択するためのグループ項目選択手段と、

サブグループ全体の縮小された表現、且つ、縮小された表現内で表示手段により現在表示されているグループの部分を示す印を表示する手段とからなり、

前記縮小された表現表示手段は3次元表現の2次元表現を表示し、3次元は、少なくともひとつの共通の属性を有する項目の論理スタックとして表現されていることを特徴とする装置。

【請求項21】 前記対話型可動式ポインティング手段は、

第1の方向へ表示内の位置における変更をコントロールするための第1の対のボタンと、第2の方向へ表示内の位置における変更をコントロールするための第2の対のボタンと、論理スタック内の位置における変化をコントロールするための第3の対のボタンとからなる遠隔制御装置を含むことを特徴とする請求項20記載の装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、プログラム（番組）の数を管理可能なグループに縮小するために多属性を有するデータ項目の多量なスケジュールを一貫した選択基準に従属させる方法および装置に関する。

【0002】

【従来の技術】現在、意志決定機能により所望の項目について視覚的に検索される管理可能なデータセットへ多量のデータを縮小するための方法がいくつか知られている。そのような多量のデータには、コンピュータシステムの固定ドライブの辞書などがある。パーソナルコンピュータのインタラクティブ・グラフィックのユーザーインターフェイスやワークステーションにより行われる方法は、一般にユーザの選択のためのルート・レベル・ファイルとひとつもしくはそれ以上の分枝サブディレクトリを示すルート・ディレクトリ表示ヘディスク・ドライ

20

3

【課題を解決するための手段】本発明の別の局面において、上述した目的は、家庭のテレビの視聴者に多属性を有するデータ項目の多量のスケジュールからあるデータ項目を対話式に選択できるようにする方法を提供することにより成し遂げられる。その方法には、データ項目の多量のスケジュールを受信する工程が含まれる。受信されたデータ項目のスケジュールは、後のフィルタリングおよび検索を促進するように局部的にあるデータベース形式に記憶される。次に、データ項目のスケジュールは、視聴者の入力に対話することのより選択された属性に従って、データ項目のスケジュールのサブグループにフィルタリングされる。その結果得られたスケジュールデータ項目のサブグループは、視聴者が確認できるように表示される。よって、ユーザは、テレビのスクリーンに移されたデータ項目のサブグループからあるデータ項

目を対話式に選択する。

【0008】要するに、本発明のひとつの局面によれば、上述した目的は、表示手段と、表示手段におけるある位置を特定し、特定された位置で選択を行うための対話型移動可能ポインティング手段とを有するシステムにおいて大きなグループから項目を選択するための装置を提供することにより成し遂げられる。

【0009】本発明の別の局面によれば、上述した目的は、視聴者が番組を対話式に選択する方法を提供することによって成し遂げられる。この方法は、少なくとも1週間の期間の少なくとも300の個々のチャンネルの番組スケジュールデータを受信する工程を含む。その受信された番組スケジュールデータは、のちの分類および検索を促進する目的でデータベースのフォーマットにローカルに蓄積される。番組スケジュールデータのサブグループは、視聴者の点検用に表示される。ユーザは、TVのスクリーン上に映したり、もしくは適切な番組レコーディング装置により録画するために番組スケジュールデータのサブグループから番組を対話式に選択する。

【0010】

【発明の実施の形態】図1を参照すると、テレビ(TV)10は、相互接続ケーブル14を介してセット・トップ・ボックス(STB)12に接続されている。STB12はまた、少なくともひとつのケーブルプログラムを搬送するケーブル16に接続されている。TV10は、たとえばNTSC方式のテレビや、高品位テレビ、その他の家庭用標準商業タイプのものなどあらゆる標準型のテレビである。コントローラ20は、確認のために番組を選択するようにSTB12の動作を制御するために、好ましくはフリースペースの光リンク22を介してSTB12にリンクされている。

【0011】図2を参照して、STB12をより詳しく記述する。STB12は、ケーブル16上で入力信号を選択し、それらがデジタル信号であろうが、アナログ信号であろうが、またはパケット信号であろうが、TV10と互換性のある信号に変換するケーブルインターフェイス30を有する。ケーブルインターフェイス30は、双方向バスによりCPU34に接続されている。双方向バス32は、CPU34で使用されるためケーブル16にわたって受信されたデジタル情報と、CPU34からケーブルインターフェイス30に伝送されたデジタル情報とを搬送する。ケーブル16が双方向ケーブルである場合、CPU34からの情報のいくつかはケーブルインターフェイス30を介してケーブル16へ処理される。

【0012】双方向バス32に加えて、CPU34はメモリバス36を介してROM38とRAM40とに接続されている。ROM38は、機能のほとんどを供給するために、CPU34により実行されるオペレーティングプログラムを含む。他のものの中で、RAM40は、CPU34によって実行されるオペレーティングプログラ

ムの中間結果のための記憶スペースを提供する。RAM40は、ケーブル16から受信され、且つオペレーティングプログラムとコントローラ20(図1に図示)からの視聴者の入力とに応じてフィルタリングされたデータの記憶機構を提供する。さらにデータのために記憶機構が必要とされる場合、より大きなRAM装置および/またはディスクドライブなどの大容量の記憶装置が双方向バス32(図示せず)に接続されてもよい。視聴者の入力を受信するには、CPU34がバス42を介してコントローラインターフェイス44に接続され、フィードバックを視聴者にていきようするには、CPU34がチャンネル関連情報とともにバス46を介してSTBディスプレイ48と接続され、STBディスプレイ48を駆動する。

【0013】図3は、コントローラ20の好ましい実施例を示す。コントローラ20は、見た目も機能もTVまたはビデオカセットレコーダ(VCR)の標準の遠隔制御装置のように設計されている。コントローラ20は、0~9の数字キーを有する数字のキーパッド50を有する。コントローラ20は、上向き矢印52、下向き矢印54、右指示矢印56、左指示矢印58、2重上向き矢印60および2重上向き矢印62とを有する。コントローラ20はまた、セレクト(レ)ボタン64、キャンセル(X)ボタン66、および問い合わせ(?)ボタン68を有する。本発明により供給されるインターフェイスでの対話はすべて、コントローラ20のこれらの19個のボタンの様々な並びにより制御される。さらに、これらのボタンのうちのひとつを動作した結果は、標準のTVやVCRの同様の動作の結果と同様であり、その使用はそれを扱う使用者にとってなじみ深く、予測がつき、且つ直観的にわかるものであろう。

【0014】本発明のインターフェイスに用いられるグラフで示した要素は、広く2つのクラスに分かれる。第1のクラスは、図4のような、視聴者が所望のデータビューを選択し、表示される情報にフィルターをかけるように用いられるものであり、第2のクラスは、図6のような、視聴者が特別な項目を選択するために進むための情報を実際に表示するために使用される要素である。たとえば、視聴者は、次の数時間の(すべてのチャンネルの)TV番組のスケジュールを確認し、表示が特別にスポーツ、バスケットボールの試合のみを示すようにフィルタリングする場合、これらの選択は、第1のクラスに属する。次の数時間についてバスケットボールの試合の表示が選択されると、視聴者は、コントローラ20により選択されたテキストもしくはビデオの要約を確認することにより特別な番組の選択は、一般的に、ゲーム番組をビデオ録音するか、所望の番組が始まることを視聴者が思い起こすようにアラームを設定するかなどの動作へ進む。所望の番組を見つけ且つ選択するための操作や選択順序は、グラフで示した要素の第2のクラスの使用の

例である。両方の場合も、視聴者が所望の番組を最終的に選択するために、複数のグラフ表示を通して操作することが要求される。インターフェイスは異なる目的を提供するものであり、視聴者はそれらの外観によって気づかされるものであるため、インターフェイスは本発明に従って概念的且つ視覚的に明瞭に保たれる。加えて、ビュー選択要素、すなわち第1のクラスに含まれた情報は、通常、階層メニューの形式、つまり複雑な基礎構造を伴う短いリストに分類される。逆に、データ表示、すなわち第2のクラスの要素は、単純な基礎構造を伴う基本的な平坦なデータである情報の多量のスケジュールおよびアレイを処理することが可能でなければならない。

【0015】加えて、現状表示として機能する「フレーム」といわれる多数の表示内の表示要素が存在する。フレームは、内容（私が表示しているのはどの画面か）とともに、現在選択されている項目の特徴（私が今選択したのはどの項目か）の要約を視聴者に知らせるために用いられる。通常、後者は、項目の名称および番組の開始時刻および終了時刻などの使用可能な情報であろう。

【0016】図3および図4を参照すると、コントローラ20で用いられる視聴者のインターフェイスの第1段階の表示400は、通常の操作中、視聴者のTV10（図1に図示）に表れる。「開始」とラベル付けされたタブを持つファイルカードメニュー402として示される。ファイルカードメニュー402には、ムービー（オンデマンド）404、最新ムービー406、オプション408、TV410、TVナウ412、最新TV414、ショッピング416および最新ショップ418とラベル付けされた対話型ボタンがあり、これらはコントローラ20により選択されたときに、次の関連した表示がRAM40（図2に図示）に記憶されたデータ上で実行される選別やフィルタリングによって示されるようになるものである。まずファイルカードメニュー402が表れているとき、選択が行われるであろう作動エリアが強調されている。この作動エリアは、コントローラ20の矢印ボタン52〜56および2重矢印ボタン60、62を作動させることにより移動可能である。ファイルカードメニュー402は、フレーム420で囲まれ、そのトップには、現在強調されている作動エリアの名称が示される。作動エリアが強調されると、図4における選択（レ）ボタン64を作動させることにより選択が行え、TVボタン410は、選択（レ）ボタン64を作動させることにより作動状態になったことが示され、図5に示された次の表示500が現れる。これは、表示400上に表示500が論理的に重なって現れる。他の論理的に重なる表示がTV10のスクリーン上に現れているときは、表示400は見えないが、論理的に重なるすべての表示がキャンセルされた場合、すなわちキャンセル

（X）ボタン66を作動することによりキャンセルされた場合、表示400はまた見えるようになる。したがっ

て、番組がリアルタイムで見えるものを選択されるまで、視聴者は、適切な回数キャンセル（X）ボタンを作動することにより表示400へ戻すことが可能である。

【0017】図5は、ラベル「開始」以外のファイルカードメニュー402のすべてに重なり塞ぐようにして現れた「TV」とラベル付けされたファイルカードメニュー502で示された第2段階の表示500を示す。ラベルTVは、アクセスできる項目が連続ドラマ、連続コメディ番組、シリーズもの、バラエティ番組、ゲーム番組、スポーツなどのTV番組であることを示している。ムービーやショッピングは他の対話型ボタンのトピックであるため、これらのタイプの番組は、全体でもしくは部分的にフィルタリングされるであろう。ファイルカードメニュー502は、オンナウ504、ウイークディ506、予告508、ウイークエンド510、検索512とラベル付けされた対話型ボタンを有する。ファイルカードメニュー402と同様に、ファイルカードメニュー502は、コントローラ20（図3に図示）の矢印ボタン52〜56および2重矢印ボタン60、62を作動させることにより視聴者が移動することができる作動エリアを有する。各作動ボタンは、選択されると実行されるであろう別のフィルタリングを示す。図5では、オンナウボタン504が強調されており、選択（レ）ボタン64を作動することにより選択された場合、図6に示された第3段階の表示が現れ、RAM40（図2に図示）に記憶されたデータのさらなる選別および／またはフィルタリングが行われる。

【0018】図6を参照すると、表示600は、現在の時刻（この図では午後6時30分）に行われているものを示す。すべてのテレビ番組の縮小された表現602は、図6に示された現在の時刻に行われているものを示す。縮小された表現602は、ぴったりと縦続接続されたカードのセット内の1枚のカードとして現在オンとなっている番組をそれぞれ提供する。そのカードは、ニュース番組、スポーツ番組、ドラマ、コメディ番組、ドキュメンタリー番組などを区別するためにグレイシェイドでコーディングされてもよい。当業者は、カラーがカラーテレビにとって好ましいことおよび縮小された表現602において区別された番組のタイプにカラーを用いる本発明による方法および装置が考えつくことが理解できるであろう。したがって、縮小された表現602内の視覚的コーディングを用いることにより、示された例においては、スポーツ番組がスポーツ以外のTV番組から目立たされている。上向き矢印52および下向き矢印54は、それぞれ、オンナウのサブグループの縮小された表現602の上下に、縮小された表現に表示された項目よりもわずかに広い選択ウインドウ604を一步步つ移動するさせる。縮小された表現602にそった作動エリアの移動は、上もしくは下の1次元である。上向き矢印60と下向き矢印62は、6ごとに減少された表現602



の上下にこの選択ウインドウ604を移動させる。選択ウインドウ604内にあって見える個々の項目は、オンナウのサブグループの縮小された表現602から6個の番組からなる更なるサブグループを示す。選択ウインドウ604のこの6個の番組のサブグループは、減少された表現602に隣り合って位置されたグリッド表示606内により大きな形式で表示される。この大きな形式により、視聴者はグリッド表示606にある番組のタイトルを読みとれるようになる。各項目の視覚的コーディングすなわちグレイシェイドのコーディングまたはカラーコーディングは、提供されるさまざまな種類の番組の違いを視聴者が区別できるようにグリッド表示606における大きな形式にも持続されている。

【0019】選択ウインドウ604およびグリッド表示606には、表示600の個々の部分におけるある項目を強調する作動エリア605、607がある。その作動エリア605、607は、上向き矢印52および下向き矢印54に応じて互いに調整されて移動する。上向き矢印52または下向き矢印54が選択ウインドウ604およびグリッド表示606よりも上もしくは下に作動エリア605および607を移動させたいとする場合、ページングが起こり、選択ウインドウが6個上がりもしくは6個下がって移動する。ある項目が作動エリア605、607内にある場合、TVのチャンネルのコールサインや、ケーブルチャンネル番号や、正確な開始および終了時刻などのさらなる情報がRAM40に記憶された番組のデータベースから検索され、表示600のフレーム610の一番上に表示される。選択(レ)ボタン64がこのとき作動されると、短いテキストの記述もしくは短い静止画または動画のビデオのいずれかのプレビューがグリッド表示606に取って代わって映り出される。これらのプレビューのためのデータはRAM40内に記憶されている。続けて2度目に選択(レ)ボタン64を作動することにより、縮小された表現602の作動エリア604における強調された番組が実際に選択される。上向き矢印52もしくは下向き矢印54が作動されると、前にプレビューされた項目の上もしくは下の次の番組の項目の各プレビューが選択される。フレーム610の一番上に表示された情報は、次の上か下の番組項目に変更する。チャンネルボタン66の作動により、表示600の前の表示に戻される。フレーム610の底部には、表示600の特徴が示され、これもRAM40から検索されるものである。問い合わせ(?)ボタン68が作動されると、生成されたヘルプメニューがグリッド表示606に取って代わって表示される。この生成されたヘルプメニューは、以下に説明するような数多くのボタンを有し、そのひとつはビューボタンである。ビューボタンが作動されると、前の選択(すなわちフィルタ)画面が生成されたヘルプメニューに取って代わって表示される。

【0020】図3および図7を参照して、カテゴリに

よる番組の選択を説明する。コントローラ20の問い合わせ(?)ボタン68の作動により、TV10(図1に図示)のスクリーンに表示700が現れる。表示700には、ヘルプボタン702、カテゴリボタン704、ビューボタン706、開始ボタン708、好み番組ボタン710およびユーザボタン712が存在する。カテゴリボタン704上に示された作動エリアは矢印ボタン52~58により移動される。ビューボタン706の機能については図6に関して説明したので、ここでは説明しない。ヘルプボタン702の作動により、開始のメニューが表示される。すなわち、選択手順の最初に戻る。好み番組ボタン710の作動により、視聴データからCPU32により蓄積されるか、またはひとりもしくは複数の視聴者により入力される現在のタイムスロットにおける好みの番組のリストが現れる。ユーザボタン712の作動により、視聴者が主要な視聴者であるとして本人のステータスに対話式に入力できる表示が現れる。この情報は、各視聴者のカスタマイズされた好みの番組のスレートを決定し、表示するのに用いられる。カテゴリボタン704の作動により、TV10のスクリーン上に表示700に取って代わって、図8に示された次の表示800が表示される。

【0021】図3、8および9を参照すると、表示800には、多数のボタン801、802、803、804、805、806、807、808および809があり、これらは、好みの番組、情報番組、エンターテインメント番組、ムービー、スポーツ番組、ニュース番組、子供番組、シリーズものおよびそれ以外のカテゴリにそれぞれ対応している。ボタン801~809は、矢印ボタン52~58を用いることにより移動される作動エリアを有し、または、数字のキーパッドが1組のホットキーとして所望のカテゴリに素早く作動エリアを移動させるために用いられる。ボタン801~809は、ちょうどキーパッド50の1~9のキーの配列と同じ縦3列横3列の配列に置かれている。したがって、ナンバリングなしでも直感でホットキーを操作することが可能である。たとえば、3番目の縦列および3番目の横列にあるボタン、すなわちボタン809、に作動エリアを移動する場合、キーパッド50の3番目の縦列および3番目の横列にあるキー、すなわち数字9のキー、を作動させる。ボタン809は、狭められた選択を連続して提供する他のボタンとは違い、さらなるカテゴリの選択(図示せず)のためのボタンの別の表示を提供する。作動エリアが図8に示されたようにスポーツボタン805にあり、選択(レ)ボタン64が作動されると、図9に示された表示が現れる。図9は、図8のものと同様に、縦3列横3列の配列の選択ボタン901~909からなる。矢印ボタン52~58もしくはホットキーによる作動エリアの移動は、図8と同様に図8においても可能である。ボタン901~909は、野球、フットボール、バ



スケートボール、サッカー、オール(全て)、ホッケー、ゴルフ、レースおよびその他にそれぞれ対応している。オールボタン905以外の図9に示されたボタンは、スポーツ全体にわたるカテゴリーの狭めたサブグループを示す。オールボタン905を作動することにより図10に示された表示1000が表示900に取って代わって、すなわち論理的に重なって表示される。

【0022】図10を参照すると、午後6時30分時点でのオンナウの、且つ、縮小された表現1002における全てのカテゴリーのスポーツ番組であるTV番組のためにフィルタリングされた表示が示されている。縮小された表現1002は、表現された完全な数の番組により前に隠された個々のカードの特徴が識別されるほど少ししかエントリを持たない。たとえば、番組「ジス・ウィーク・イン・ザNBA」を示すメニューカード1003は、番組「シニアPGAゴルフ」を示すメニューカードよりも左側が短くなっている。これは、番組「ジス・ウィーク・イン・ザNBA」が午後6:30から始まり、一方、番組「シニアPGAゴルフ」は、選択ウインドウ1006における「シニアPGAゴルフ」のタイトルの前の2重左指示矢印によって示されるようにより早い時間から始まる。「ジス・ウィーク・イン・ザNBA」と「シニアPGAゴルフ」とは両方とも同じ時刻に終了するので、縮小された表現1003および1004の右側は、同じ位置で終わっている。縮小された表現1002の作動エリア1005の移動もしくは操作は、図6と同様にコントローラによって行う。選択ウインドウ1006に示された6つのタイトルはそれぞれ、長方形領域1010~1015を持つ。長方形領域1010~1015は、それらが属するスポーツ番組の種類によって異な

って陰付け(シェーディング)がされている。これらの異なったグレイの陰もしくは表示がカラーテレビで映される場合異なったカラーの陰は、6つのタイトルそれぞれに対応するスポーツの種類に対する視覚的キーである。コントローラ20の選択(レ)ボタン64を作動することにより、作動エリアにより強調されたプログラムのテキストプレビューである図11に示された表示が選択エリア1006に重なって表示される。上述したように、選択(レ)ボタン64は、この時点で、CPUが、確認のためそのテレビ番組を選択するようケーブルインターフェイス30(図2に図示)に指示するようにする。

【0023】再度図10を参照すると、方法および装置の状態が表示1000が表示されるようにする選択が行われた直後のそれと同じである場合、且つ、前のパラグラフで説明したように問い合わせ(?)ボタン68が作動された場合、種々の選択とともに図7に示された表示700が再度表示される。さらに、ビューボタン706が作動された場合、図12に示された表示1200およびそのフィルタ選択が表示700上に論理的に重なる。

表示1200は、オンナウ1202、予告1204、検索1206、ウィークデー1208およびウィークエンド1210の対話型ボタンを有する。全てのスポーツのカテゴリーが前もって選択されたので、表示1200の作動エリアが予告ボタン1204を強調するように移動されて、ボタン1204が作動されると、図14に示された表示1300が現れ、表示1200上に重なる。

【0024】表示1300において、データ項目の選択されたサブグループの2つの属性でそれぞれラベル付けされた2つの座標軸が示されている。表示1300に示された2つの属性は、次の24時間のタイムスロットつまり12時間のためのチャンネルとタイムスロットである。すべてのスポーツカテゴリーが選択されているため、次の12時間以内の300以上のチャンネルに示されるスポーツ番組がそれぞれ表示1300に示されるであろう。各スポーツ番組の予告は、その番組を搬送するチャンネルに対応する横の列に且つその番組が放映される時刻を示す縦の列に位置づけされた矩形の「カード」によって示される。視聴者は、垂直および水平方向それぞれの移動のために上および下向き矢印52、54および右および左指示矢印56、58を用いてどのカードにも作動エリア1302を移動させることができる。表示1300からわかるように、常識的な時間内でそれぞれを考慮するにはサブグループにはまだ多くのデータ項目があるので、より短い時間すなわちオンナウのもの、または、より狭められたカテゴリーすなわちバスケットボールによるさらなるフィルタリングが必要とされる。より狭められたカテゴリーを変更するには、視聴者は、問い合わせ(?)ボタン68を押し、表示700(図7に図示)を表示する。次にカテゴリーボタン704を選択し、表示900(図9に図示)を表示する。次に、バスケットボールボタン903を選択し、図15の表示1500を表示する。図12及び13の予告時間フィルタは変更しないので、表示1500は次の12時間に放映されるバスケットボールの番組を示す。図から分かるように、2次元のグリッド表示1500は、約16の番組を含んでいる。この数は常識的な時間内でそれぞれを確認するのに十分に少ない数である。上および下向き矢印52、54および/または右および左向き矢印56、58を用いて2次元のグリッド表示1500内で作動エリア1500を移動させることにより、確認および選択工程を助けるために各番組のタイトルやチャンネルを表示1500のフレームの一番上に表示させる。たとえば、作動エリア1502により強調された番組は、「ジス・ウィーク・イン・ザNBA」であり、それはCNNで放映される。したがって、図13に示された扱いにくい番組の表示1300を選択的にフィルタリングすることによって、管理可能な少数の表示1500に縮小することができ、視聴者は常識的な時間内でそれぞれを操作することができる。

【0025】図16～23を参照して、本発明の別の局面を説明する。図16および残りの図面では、1時間半よりも長時間が選択され、オンナウ選択により検索される。たとえば、視聴者がVCR（図示せず）に録画するものを選択するために今週の残りにおける可能なプログラムについて見たいとする場合、番組を見ているときキーパッド50の数字0のボタンを作用することにより、最終選択を行うメニュー画面の階層で図9の900のようなデータビューメニュー選択カードがこのとき現れる。再度0のボタンを作用すると、図4の最も広いデータビューメニュー400に移動され、視聴者は時刻もしくは主題のカテゴリーを変更するためのいかなる表示でも停止することができる。

【0026】よって、視聴者が「ジス・ウィーク・イン・ザNBA」を見ていて、その後の興味ある番組を見たいとした場合、視聴者は、まずキーパッド50の0ボタンを作用し、図10の表示を出すであろう。4回以上0のボタンを押すと、図9、8、7および5それぞれの表示900、800、700および500へと進む。特定の番組のタイトルを出すには、検索ボタン509を作用させ、図16を表示500に論理的に重ねる。図16は、対話型英数字表示の選択シーケンスの第1表示1600を示す。まず、すべての英数字表示のタイトルは、5つもしくはそれ以下のグループに分類される。たとえば、「NOVA」が所望の番組のタイトルであった場合、作動エリアは、選択（レ）ボタン64の作用により追従される図17に示されたように上向き矢印52または下向き矢印54を用いてその初期位置（表示の一番上か選択された最後のグループのどちらか）から文字Nを含む文字のグループに移動される。このシーケンスにより、図18が図17の上に論理的に重なる。図18において、作動エリアは、図17に示されたようにMにおける初期位置からNの位置へ移動されて、再度選択（レ）ボタン64の作用により図19上に図20が重なるようになる。表示2000には、最初の2つの文字についての実例が示されている。たとえば、NYPDブルーは、YがNの後に続くたったひとつの例である。そして、2つの文字の列について複数の例があるとき右指示矢印によってNOにより示される。NOVAについて検索をつづけるには、下向き矢印56を用いて作動エリアを図21に示されたように表示2000のNOを含むラインに移動し、選択（レ）ボタン64を作用することにより、図22の表示2200が表示2000に重なる。現在、NOVAは、NOVで始まる番組のたったひとつの例であるので、完全なタイトルNOVAが図22に現れる。表示2200においてNOVAとラベル付けされたラインに作動エリアを移動し、選択（レ）ボタン64を作用することにより、シリーズ番組NOVAの時刻表とチャンネルとともに図23に示された表示2300が表示2200上に重なる。

【0027】図23は、論理的3次元のグリッドとして構築された1週間のスケジュールである。その週の曜日は、表示2300の一方の側に、この場合は縦に且つ左側に、表示されている。1日の時刻はそれと垂直をなす側、この場合は一番上に水平方向にそって24時間で表示されている。よって、NOVAのエピソードが日曜日の午後8時に予定されている場合、陰付きのボックスは日曜日の列と午後8時の列とが交わったところに配されている。作動エリア2302は、キーパッド50の矢印56、58で水平方向に、矢印52、54で垂直方向に移動可能である。特定の夜の特定の時刻にNOVAが複数放映される場合、ボックス内に星印（\*）を伴って、その日の列とその時刻の列とが交わったところに配されたボックスにより示される。星印（\*）は、水曜日の夜8時に放映されるなど、競合するチャンネルに現れるNOVAの複数のプログラムの論理スタックの存在を示す。特定の日の特定の時刻における番組のスタック（たとえば、同じ名称の番組のエピソードのスタック）を介して移動もしくは操作するには、この第3段階の自由のために二重上向き矢印ボタン60および二重下向き矢印ボタン62を使用する。表示2300が番組タイトルはもちろんとして視覚的な識別をより必要とすることもあるので、フレームの情報ウィンドウ1904は表示2300については通常より大きい。さらに、フレーム2304は、スタック内の作動エリア位置より上もしくは下にある番組エピソードの存在を示す矢印により注釈されている。ケーブル16がプログラムの300以上の「チャンネル」へのアクセスを有する場合、NOVAなどのいくつかのプログラムが同じ時間にひとつ以上のチャンネルにより提供されることもあり得る。前述したように、視聴者が2次元もしくは3次元において特定のエントリに作動エリアを移動し、選択（レ）ボタン64を作用すると、選択が行われる。この場合、選択は、数日以内の特定の時間に特定のチャンネルを記憶するためのアラームを設定する。

【0028】図1および2に戻り、本発明の装置の全体の操作について記述する。番組スケジュールのデータはケーブル16を介して供給される。この番組スケジュールのデータは定期的に送信され、STB12はこの番組スケジュールデータを受信し、RAM40に記憶する。あるいは、その番組スケジュールデータの全てもしくは一部は、動的に要求され、STB12により受信され、RAM40に記憶される。このような番組データは、TV Data, Inc. およびその他の同様の会社から商業的に入手可能である。番組スケジュールデータのデータもしくはレコードは、CPU34による記憶、分類および検索を速めるために、マイクロソフト・アクセスやその他の同様のデータベースフォーマットなどの前もってアレンジされたフォーマットとされている。TV番組の各レコードは、放映の日付、放映時間、タイトル、

チャンネルおよび／またはネットワーク、種別、およびテキストによるまたは視覚的なプレビュー（もしあれば）を有する。分類し、表示し、視聴者の入力に対話的に答えるプロトタイププログラムのリストは、以下のCPUプログラムリストに示されている。これは、マイクロソフト社のビジュアル・ベーシック・プログラム言語でリストにされている。

【0029】ビジュアル・ベーシック・プロトタイプ・プログラムは、それぞれ事象操作子の固有のセットを有する形式の集合からなる。この場合、遠隔制御装置のため、重要な外部事象のみがボタン操作である。フレームフォームは、ほとんどの個々の表示に用いられるバックグラウンドおよび情報およびステータスバーを提供する。ローロデックスフォーム（rolodex form）は、メニューを提供する。その他のフォームは、通常は、英数字の選択リストフォームなどの専門的な種類を含む様々な種類のスケジュールもしくはリスト表示で

ある。

【0030】プログラムの制御部は、すべてのフォームをロードする手続きから開始し、フレームおよびローロデックス、すなわちトップディスプレイ、を起動させる。フォームは、リターンコードを設定したり、それら自身を隠したりして制御をハンドオフする。これにより（通常フレームの）直接下にフォームを起動させる。フレームフォームとローロデックスフォームは両方とも、リターンコードの値によって異なる動作を行う。フレームフォームの最も共通の動作は、別のフォームを軌道することであり、アプリケーションの制御フローのほとんどはフレームフォームコードにより処理されている。ローロデックスフォームは、いくつかの異なるメニュー階層、最も重要な画面選択およびフィルター選択を表示するのに用いられる。

【0031】

19

20

'===== COMING form code ====='

この形式は、一日のうちの数時間のテレビ番組スケジュールを表示する。

このバージョンは、番組形状の引き出し方法を用いる

(各番組のコントロール形状を創造するものと対抗するものとして)

および'ポイント&ショット'もしくは'視覚的クローゼット'操作

Option Explicit

Dim allData(8) As snapshot 時間内のすべてのデータ

Dim filterData(8) As snapshot 日ごとの確認のスナップショット

Dim NDays As Integer 表示内の日数

Dim NSlots As Integer 表示内の時間数

Dim NStation As Integer 表示内の局数

Dim MaxStation As Integer データベース内の総局数

Dim colorField As String 項目のカラーを決定するデータベースフィールド

(フィールドは整数を含む)

Dim inPreview As Integer 'boolean プレビューメッセージは示すべきか?

Const sideGap = .05 番組の開始と終了の間隔

Const topGAP = 4 タイムラベルと第1の番組形状との間隔

Dim refDate As Integer データ時刻のリファレンスデート

Const lblHeight = 40 日付および時間ラベルの高さ(500スケール)

Const MINProgWidth = .2 スロットの番組形状の最小幅

Dim slotsPerDay As Integer 一日に許容されるスロットの数

Dim currDay As Integer 現在の日付

Dim startTime As Integer 表示の開始日時

Dim TSBegin As Long 最初の時刻

Dim TSEnd As Long 最後の時刻

Dim TScurrent As Long 現在の時刻

Dim rowOffset As Integer スケジュールの列(上列)間の距離

Sub ApplyFilter ()

番組データをフィルタリング、フィルタ(TV)内の問い合わせに合致する番組のみをキーブ

局数が正しいかを確認

カラーを決定するDBフィールドが設定される

Dim i As Integer カウンタ

If InStr(filters(currDomain), "Station") Then

NStation = 10 注:これはおそらく10以上の変数もしくは定数であるべきである

colorField = "Type"

Else

NStation = MaxStation

colorField = "Category"

End If

For i = 1 To NDays

allData(i).Filter = filters(currDomain)

Set filterData(i) = allData(i).CreateSnapshot()

Next i

End Sub

Sub ChangeSel (id As String)

方向パラメータに従って操作を行う

Dim current, firstMatch データベースメーカー

Dim success As Integer 論理値(ブール)

Dim s As Integer 局(ステーション)

21

22

```

Dim e          TS終了
Dim TS As Long   タイムスロット
Dim F As snapshot
Dim aDay As Integer
Dim dist 距離
Dim best As Long, bestMark データベースマーカーとして

データベース内の現在地についての情報をセット
current = filterData(currDay).Bookmark
Set F = filterData(currDay)
s = F("Station")
e = F("FinishTS")
TS = TScurrent
aDay = currDay
success = False
best = 9999

If d = "Right" Then
    即座に右をチェック
    F.MoveNext
    If Not F.EOF Then
        TS = F("StartTS")
        成功=同じステーションそして現在の番組後に右を開始
        success = (F("Station") = s) And (TS <= e + 1)
    End If
    If Not success Then
        'closest'の右に対してすべてをチェック
        F.MoveFirst
        While Not F.EOF
            If F("FinishTS") > e Then
                dist = VDistHoriz(s, e, F("Station"), F("StartTS"))
                If dist <= best Then
                    今のところの最善を保つ
                    best = dist
                    success = True
                    bestMark = F.Bookmark
                End If
            End If
            F.MoveNext
        Wend
        If success Then
            最良のものに移動
            F.Bookmark = bestMark
            TS = F("StartTS")
        End If
    End If
ElseIf d = "Left" Then
    即座に左をチェック
    F.MovePrevious
    If Not F.BOF Then
        成功=同じステーションそして現在の番組の前に右を終了
        success = (F("Station") = s) And (F("FinishTS") >= TS - 1)
        TS = F("StartTS")
    End If

```

23

24

```

If Not success Then
  'closest'について左に對してすべてをチェック
  F.MoveFirst
  While Not F.EOF
    If F("StartTS") < TScurrent Then
      dist = VDistHoriz(F("Station"), F("FinishTS"), s, TScurrent)
      If dist < best Then
        これまでのベストをキープ
        best = dist
        success = True
        bestMark = F.Bookmark
      End If
    End If
    F.MoveNext
  Wend
  If success Then
    最良のものへ移動
    F.Bookmark = bestMark
    TS = F("StartTS")
  End If
End If
ElseIf d = "Down" Then
  'closest'を保ったまま現在のもの以下のすべての番組をチェック
  While Not F.EOF
    If F("Station") > s Then
      dist = VDistVert(s, TScurrent, e, F("Station"), F("StartTS"),
F("FinishTS"))
      If dist < best Then
        best = dist
        success = True
        bestMark = F.Bookmark
      End If
    End If
    F.MoveNext
  Wend
  If success Then
    F.Bookmark = bestMark
    TS = F("StartTS")
  End If
ElseIf d = "Up" Then
  'closest'をキープして現在のものより上のすべての番組をチェック
  While Not F.BOF
    If F("Station") < s Then
      dist = VDistVert(s, TScurrent, e, F("Station"), F("StartTS"),
F("FinishTS"))
      If dist < best Then
        best = dist
        success = True
        bestMark = F.Bookmark
      End If
    End If
    F.MovePrevious
  Wend
  If success Then

```

```

25
    F.Bookmark = bestMark
    TS = F("StartTS")
End If
End If

If success Then
    盒蓋および表示を更新
    TScurrent = TS
    currDay = aDay
    DisplayProg
Else
    データベース内の古いポジションを回復
    filterData(currDay).Bookmark = current
End If
End Sub

Sub DisplayProg ()
    情報ボックス内に現在の番組の情報を設定
    画面内の適切な番組形状を強調
    Dim F As snapshot
    Dim msg As String

    Set F = filterData(currDay)

    強調を設定
    shpProg(0).Visible = False
    selector.Visible = False
    Position shpProg(0), F("Start"), F("Finish"), F("Station")
    CPlace 0, selector, shpProg(0)
    shpProg(0).Visible = True
    selector.Visible = True

    情報ボックスのためメッセージ
    msg = StationString(F("Station")) & " - " & F("Title") & " "
    msg = msg & Format(F("Start"), "h:mm AM/PM")
    msg = msg & " to " & Format(F("Finish"), "h:mm AM/PM")
    SetInfo msg, Color(F(colorField) Mod 9)
End Sub

Sub DoPreview ()
    適切なプレビューメッセージを構築そして表示
    Dim msg As String

    msg = "Station: " & StationString(filterData(currDay)("Station"))
    msg = msg & Chr(13) & "Title: " & filterData(currDay)("Title") & Chr(13)
    msg = msg & CategoryString((filterData(currDay)("Type")),
(filterData(currDay)("Category")))
    msg = msg & Chr(13) & "Time: " & Format(filterData(currDay)("Start"), "mmm d,yy
h:mm AM/PM")
    msg = msg & Chr(13) & "    to " & Format(filterData(currDay)("Finish"), "h:mm
AM/PM")

    プレビューメッセージとともにポップアップをバす
    popup.Caption = msg

```



27

28

```

popup.Top = lblTime(1).Top + 2 * lblTime(1).Height
popup.Left = 2
popup.Width = slotsPerDay - 3
popup.Visible = True
inPreview = True
End Sub

Sub DoSelect ()
    選択情報をセットしTVへ行く
    userStation = filterData(currDay)("Station")
    userStart = filterData(currDay)("Start")
    returnCode = TOTV
    Me.Hide
End Sub

Sub DrawProg (colorIndex, start, finish, station)
    番組形状を引き出すために引き出し方法を使用
    'note: form.AutoRedraw should be set to true so the drawings are persistent
    Dim L, R, t, B 左, 右, トップ, ボトム
    Dim dayStart
    Dim edge

    NSlotスケールのポジションへ日付/時刻を交換
    dayStart = startTime + currDay - 1
    L = (start - dayStart) * 48
    R = (finish - dayStart) * 48
    日付け境界において形状を切り取る
    If L < 0 Then L = 0
    If R > slotsPerDay Then R = slotsPerDay
    番組間の小さなすき間を正しい日に置く
    edge = (currDay - 1) * slotsPerDay
    L = L + edge + sideGap
    R = R + edge - sideGap
    番組が出ることを確認するための最小幅を訂正
    If R - L < MINProgWidth Then R = L + MINProgWidth
    局によってトップをセット
    注: このトリックは'好み'の局が1..nの符号がつけられていない場合には機能しない
    rowOffset = ((500 - 2 * lblHeight - shpProg(0).Height) / NStation)
    t = shpSlot(0).Top + topGAP + (station - 1) * rowOffset
    B = t + shpProg(0).Height
    正しいカラーでボックスを引く
    drawwidth = 1
    Me.FillStyle = 0 ソリッド
    Me.FillColor = Color(colorIndex Mod 9)
    Line (L, t)-Step(R - L, B - t), B 論的Bを伴うラインコマンドがボックスを引く
box
End Sub

Sub Form_Activate ()
    表示に必要な変更を行い、情報およびステータスバーをリセット
    Dim i As Integer カウンタ
    Static saveFilter As String

    If saveFilter = filters(currDomain) Then sameFilter = True

```

29

30

```

saveFilter = filters(currDomain)
setStatus "TV Coming Up: " & currFilter(TV), greyCOLOR
If newUser Then
    popup.Caption = "Press 'カテゴリ'を押して表示されている番組の種類を変更"
    popup.Visible = True
    newUser = False
End If

同じフィルタでない場合、表示をredo (設計し直す(?))
If Not sameFilter Then
    SetInfo "Loading program information...", GREY
    shpProg(0).Visible = False
    selector.Visible = False
    ApplyFilter
    MakeDisplay
End If

すべての場合において
DisplayProg
If inPreview Then DoPreview
End Sub

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)
    Select Case KeyCode
        Case Asc("Q")
            End
        Case B_BACK
            returnCode = BACK
            Me.Hide
        Case B_HELP
            InvokeHelp
        Case B_PREVIEW
            If inPreview Then
                popup.Visible = False
                inPreview = False
            Else
                inPreview = True
            End If
        Case B_RIGHT
            If Not filterData(currDay).EOF Then ChangeSel ("Right")
        Case B_LEFT
            If Not filterData(currDay).EOF Then ChangeSel ("Left")
        Case B_UP
            If Not filterData(currDay).EOF Then ChangeSel ("Up")
        Case B_DOWN
            If Not filterData(currDay).EOF Then ChangeSel ("Down")
        Case B_SELECT
            If Not filterData(currDay).EOF Then DoSelect
        Case B_PAGEDOWN
        Case B_PAGEUP
        Case B_FILTER
            returnCode = Filter
            Me.Hide
        Case B_0
    
```

31

32

```

        returnCode = SHORTCUT
        Me.Hide
    End Select

    すべての場合において
    If inPreview Then
        DoPreview
    Else
        popup.Visible = False
    End If
End Sub

Sub Form_Load ()
    Dim i As Integer
    Dim t 'as time

    フォームカラーとフォントをセット
    Me.BackColor = formCOLOR
    shpProg(0).BackColor = BorderColor
    lblDay(0).BackColor = backgroundCOLOR
    lblAM.BackColor = backgroundCOLOR
    lblPM.BackColor = backgroundCOLOR
    selector.BorderColor = BorderColor
    dayLine(0).BorderColor = divideColor
    lblTime(0).ForeColor = slotCOLOR
    shpSlot(0).BorderColor = slotCOLOR
    If displayMode = 'TV' Then
        lblDay(0).FontSize = smallFONT
        lblTime(0).FontSize = smallFONT
        lblAM.FontSize = smallFONT
        lblPM.FontSize = smallFONT
        popup.FontSize = mediumFONT
    Else
        lblDay(0).FontSize = largeFONT
        lblTime(0).FontSize = largeFONT
        lblAM.FontSize = largeFONT
        lblPM.FontSize = largeFONT
        popup.FontSize = largeFONT
    End If

    スケールおよびサイズオブジェクトをセット
    SizeAForm Me, DispTop, DispHeight, DispLeft, DispWidth
    Me.Scale (0, 0)-(500, 500)
    SizeAControl lblDay(0), 0, lblHeight, 0, 500
    注: AM/PMラベルは時間がフィルタリングされた場合に置かれる
    SizeAControl lblPM, 0, lblHeight, 0, 30
    SizeAControl lblAM, 0, lblHeight, 500 - 30, 30
    SizeAControl lblTime(0), lblHeight, lblHeight, 0, 50
    SizeAControl shpSlot(0), 2 * lblHeight + .5 * topGAP, 500 - 2 * lblHeight, 0, 50
    SizeAControl popup, 250, 200, 250, 200
    selector.BorderWidth = 1
    dayLine(0).Y1 = 0
    dnyLine(0).Y2 = 500
    変数をインシャライズ

```

33

34

```

    startTime = fakeToday + fakeTime 'this would be set at activate to current half
hour
    NDays = 1
    slotsPerDay = 24
    NSlots = NDays * slotsPerDay
    sameFilter = False
    sameView = False
    inPreview = False
    フォームスケールおよびプレイスバロメント要素 (day and time labels)
    Me.ScaleWidth = NSlots
    Load lblDay(1)
    SizeAControl lblDay(1), 0, lblHeight, 0, slotsPerDay
    lblDay(1).Caption = DayString(startTime, "long")
    lblDay(1).Visible = True
    lblTime(0).Width = 1
    For i = 1 To slotsPerDay
        Load lblTime(i)
        lblTime(i).Move i - 1
        t = DateAdd("n", 30 * (i - 1), startTime) 30分の間隔を加える
        lblTime(i).Caption = TimeLabel(t)
        lblTime(i).Visible = True
        lblTime(i).ZOrder
    Next i

    InputData
    Form_Activate
    sameView = True
End Sub

Sub InputData ()
    forw_loadの1部
    データベースを開け

    Dim DB As database
    Dim RefSnap As snapshot

    Set DB = OpenDatabase(TVDB)

    'get reference date and number of stations
    Set RefSnap = DB.CreateSnapshot("Reference")
    RefSnap.FindFirst "名称='日付'"
    refDate = DateValue(RefSnap("Data"))
    RefSnap.FindFirst "名称='NSステーション'"
    MaxStation = Val(RefSnap("Data"))

    Set allData(0) = DB.CreateSnapshot("Programs")
    データはすでに分類されたものと仮定

    特別な期間のためのフィルタ、30分おきに変化がおこる
    TSBegin = Abs(DateDiff("n", startTime, refDate) \ 30)
    TSEnd = TSBegin + slotsPerDay - 1 slotsPerDayがセットされたことをチェック
    allData(0).Filter = Overlap(TSBegin, TSEnd)
    Set allData(1) = allData(0).CreateSnapshot()
    Set allData(0) = Nothing 'won't be needing everything

```

35

36

End Sub

Sub MakeDisplay ()

データから番組の表示を割り出す

Dim i As Integer カウンタ

Dim d As Integer 日にち

Dim F As Snapshot 区画

If Not sameView Then

時間と日にちの記述をリセットすることが必要か

End If

番組形状を置く

Cls プレビュー引き出しの形状をクリアする

DoEvents そのようにする

For d = 1 To NDays

currDay = d

時刻を削けるためラインを引く

For i = 0 To slotsPerDay

drawwidth = 4

Line (i, shpSlot(0).Top)-(i, 500), slotCOLOR

Next i

データにおける各番組の番組形状を引く

Set P = filterData(d)

If Not P.EOF Then

F.MoveFirst

Do While Not F.EOF

DrawProg F(colorField), F("Start"), F("Finish"), F("Station")

F.MoveNext

Loop

F.MoveFirst

End If

Next d

表示を初期化

TSCurrent = TSBegin

currDay = 1

shpProg(0).ZOrder

selector.ZOrder

Set F = filterData(currDay)

開始のため番組を見つける

Do While TSCurrent &lt;= TSEnd

F.FindFirst Overlap(TSCurrent, TSCurrent)

If Not F.NoMatch Then

DisplayProg

Exit Do

End If

TSCurrent = TSCurrent + 1

Loop

TSCurrentが範囲内にあることを確認

If TSCurrent &gt; TSEnd Then TSCurrent = TSBegin

End Sub

Sub Position (shape As Control, start, finish, station)

番組形状コントロールを位置づけ

```

Dim relativeL, relativeW, dayStart
Dim edge
日付/時間をNSlotスケールの位置に算換
dayStart = startTime + currDay - 1
relativeL = (start - dayStart) * 48
relativeW = (finish - dayStart) * 48 - relativeL
日付け境界において形状を切り取る
If relativeL < 0 Then
    relativeW = relativeW + relativeL
    relativeL = 0
End If
If relativeW + relativeL > slotsPerDay Then relativeW = slotsPerDay - relativeL
番組間の小さなすき間をそのままにして形状の左右および幅をセット
edge = (currDay - 1) * slotsPerDay
shape.Left = relativeL + edge + sideGap
shape.Width = relativeW - 2 * sideGap
番組が見れるように最小幅をセット
If shape.Width < MINProgWidth Then shape.Width = MINProgWidth
ステーションに従ってトップをセット
注: '好み'ステーションが1..nと符号が付けられていない場合はこれは機能しない
rowOffset = ((500 - 2 * lblHeight - shpProg(0).Height) / NStation)
shape.Top = shpSlot(0).Top + topGAP + (station - 1) * rowOffset
End Sub

```

Function VDistHoriz (station1, finish, station2, start)

2つの番組間の'視覚的'な左から右の距離の値を計算

より早い番組が最初にくることを要求

注: 微調整必要、特にぎっしりと詰まった表示だと十分に機能しない

```

Dim deltaR, deltaT 'change in row and time
Dim row1, row2

row1 = station1
row2 = station2
注: ステーションが1..nと符号が付けられていない場合は列の計算が更に複雑になる
deltaR = Abs(row1 - row2) * (100 / NStation)
deltaT = (start - finish) * (100 / slotsPerDay)
'penalize programs that are more up/down than to side
If deltaT <= 1 Then deltaT = (finish + 1 - start) * (100 / slotsPerDay)
If deltaT < 1 Then deltaT = 100 / slotsPerDay 'don't allow zero
VDistHoriz = deltaR + deltaT
End Function

```

Function VDistVert (station1, start1, finish1, station2, start2, finish2)

2つの番組間の上下の距離の値を計算

注: 微調整必要

```

Dim deltaR, deltaT 'change in row and time
Dim row1, row2

row1 = station1
row2 = station2
注: ステーションが1..nと符号が付けられていない場合は列の計算が更に複雑になる
deltaR = Abs(row1 - row2) / NStation

```

If start1 > finish2 Then [0032]

deltaT = Abs(start1 - finish2)

ElseIf start2 > finish1 Then

deltaT = Abs(start2 - finish1)

Else

deltaT = 0

40

End If

VDistVert = deltaR + 2 \* deltaT

End Function

39

40

```

'====フレームフォームコード====
このフォームは標準情報およびステータスバーを所有し、フォームからフォームへの
コントロールの伝送を許可する
Option Explicit

Sub Form_Activate ()
  他のフォームが表示エリアにおいて示すべきことを決定
  Select Case returnCode
    Case SHOWVIEW
      views(currDomain).Show
    Case PICK
      frmSelect.Show
    Case TOTV
      frmTV.Show
    Case LASTVIEW
      sameFilter = True
      views(currDomain).Show
    Case STARTUP
      なにもしない--まだローロデックスに示してもらいたいと思わない
    Case Else
      frmDex.Show
  End Select
End Sub

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)
  If KeyCode = Asc("Q") Then
    End
  End If
End Sub

Sub Form_Load ()
  カラーとフォントをセット
  Me.BackColor = formCOLOR
  sspInfo.FontSize = mediumFONT
  sspStatus.FontSize = mediumFONT
  バックグラウンドのサイズをあわせる為builtinオブジェクトを使用
  ScrWidth = Screen.Width
  ScrHeight = Screen.Height
  If displayMode = "mini" Then
    スクリーンのプリントをとるために
    ScrHeight = ScrHeight * .54
    ScrWidth = ScrWidth * .712
    displayMode = "TV"
  Else
    TVに合わせるためにサイドサイズを合む
    ScrHeight = ScrHeight * .83
  End If
  スクリーンをうめるようにフォームをセット
  frmFrame.Top = 0
  frmFrame.Height = ScrHeight
  frmFrame.Left = 0
  frmFrame.Width = ScrWidth
  スクリーンのトップの情報ライン
  sspInfo.Visible = True
  スクリーンのボトムのステータスライン
  sspStatus.Visible = True
  利用可能な表示エリアを測定
  DispTop = sspInfo.Height + 1.5
  DispHeight = frmFrame.Height - (sspStatus.Height + 1.5)
  DispLeft = 0
  DispWidth = frmFrame.Width
End Sub

Sub SetupStatus ()
End Sub

```

【0033】



41

42

```

'====リストフォームコード====
このコードは3つのリストのフォーム（TV、ムービー、ショッピング）すべてに使用される
Option Explicit
Dim DB As database 'full database with indexes

Dim BlinkControl As Control 明滅するオブジェクト（現在はなし）にセット
Dim itemSelected As Integer 1からMAXDISPLAYへ
Dim locSelected As Integer 1からMAXLOCへ
Dim inPreview As Integer 論理値（ブール）
Dim captionField As String 表示に用いられるデータベースフィールド
Dim startTime As Long TVリストの開始時間
Dim TS As Long TVリストのタイムスロット
Dim rowOffset As Integer 2つの連続的縮小した項目のトップ間の差
Dim browsing As String 現在のショッピングリストのタイプ
Dim colorField As String カラーを決定するフィールド（タイプ整数となるべき）

表示パラメータ
Const MAXDISPLAY = 6 クローズアップの項目数
Dim MAXITEM As Integer 全リストにおける項目数
Dim MAXLOC As Integer ロケータ位置の数
Dim whichrItem(MAXDISPLAY) As Integer rItemは現在表示にある

ロケータおよびセレクトのサイズを固定
Const GAP = 10 リストまわりのスペース
Const EXTRA = 70 長めの番組のためのルーム
Const reducedEXTRA = 20 縮小した表現における長めの番組のためのルーム
Const T = 50 縮小されたリスト
Const H = 1000 - 2 * T
Const locL = 30 表示エリア
Const locW = 100
Const displ = locW + 2 * locL
Const dispW = 1000 - displ - locL

```

30

40

50

43

44

```

データベース・スナップショット
Dim allData As snapshot
Dim itemData As snapshot
Dim storeData As snapshot
Dim deptData As snapshot
Dim stuffData As snapshot
Dim filterData As snapshot
Dim marker(1000) As String 各MAXDISPLAY項目ごとのブックマーク
Dim locStart(1000) 'rItem index for start of locator

Sub ApplyFilter ()
  ユーザの選択に従ってデータをフィルタリング
  Dim sortString As String

  If Me Is TVlist Then
    captionField = "Title"
    sortString = ""
    colorField = "Type"
  ElseIf Me Is MOVlist Then
    If sameView Then
      allDataをあるがままにキープ
    Else
      allDataをすべてのムービーへリセット
      LoadData
      allData.Filter = viewFilter
      Set allData = allData.CreateSnapshot()
    End If
    captionField = "Title"
    sortString = "Title"
    colorField = "Type"
  ElseIf Me Is SHOPlist Then
    注：これはすべて全体として異なって終わるであろう。
    それを理解する邪魔をせず、ただリライトせよ
    Select Case filters(currDomain)
    Case "store"
      browsing = "store"
      Set allData = storeData
      captionField = "name"
      filters(currDomain) = ""
      sortString = "name"
      colorField = ""
    Case "dept"
      browsing = "dept"
      deptData.FindFirst "name = " & userString & ""
      userString = "" 'fix--this is cheating, userStringは使用すべきでない
      If deptData.NoMatch Then
        Set allData = deptData
        filters(currDomain) = ""
      Else
        browsing = "stuff"
        filters(currDomain) = "[dept code] = " & deptData("code")
        Set allData = stuffData
      End If
    End Select
  End Sub

```

40

45

46

```

captionField = "name"
sortString = "name"
colorField = ""
Case "item"
    browsing = "item"
    Set allData = itemData
    captionField = "name"
    colorField = ""
    filters(currDomain) = "name like " & userString & ""
    sortString = "name"
Case Else
    browsing = "stuff"
    captionField = "name"
    sortString = "name"
    colorField = "[item code]"
    Set allData = stuffData
End Select
End If

allData.Filter = filters(currDomain)
If filters(currDomain) = "" Then
    allData.FindFirst "Not " & captionField & " = ""
Else
    allData.FindFirst allData.Filter
End If
If allData.NoMatch Then
    MAXITEM = 0
Else
    MAXITEM = 1  単にそれが0でないことを確かめるため一次的にセット
    Set filterData = allData.CreateSnapshot()
    filterData.Sort = sortString
    Set filterData = filterData.CreateSnapshot()
End If
End Sub

Sub BlinkStart (C As Control, vis)
    Set BlinkControl = C
    BlinkControl.Visible = vis
    tmrBlink.Enabled = True
End Sub

Sub BlinkStop (vis)
    tmrBlink.Enabled = False
    If BlinkControl Is Nothing Then
        なにもしない
    Else
        BlinkControl.Visible = True
    End If
    Set BlinkControl = Nothing
End Sub

Sub ChangeLoc (direct As String)
    ロケータでページをめくるまたは戻す
    Select Case direct

```

```

47
Case "Up"
  If locSelected > 1 Then
    locSelected = locSelected - 1
    RedoDisplay
  End If
Case "Down"
  If locSelected < MAXLOC Then
    locSelected = locSelected + 1
    RedoDisplay
  End If
End Select
End Sub

Sub ChangeSel (direct As String)
  ある選択を上下に操作
  Select Case direct
    Case "Up"
      If itemSelected > 1 Then
        現在の表示内で上に移動
        itemSelected = itemSelected - 1
        selector.Top = itemBox(itemSelected).Top - GAP
        rItem(0).Top = rItem(whichrItem(itemSelected)).Top
        rItem(0).Left = locL - GAP
        rItem(0).Width = locW + 2 * GAP
        SetItemInfo
      ElseIf locSelected > 1 Then
        リストの前のセクションを表示
        itemSelected = MAXDISPLAY
        locSelected = locSelected - 1
        RedoDisplay
      End If
    Case "Down"
      If itemSelected < MAXDISPLAY Then
        現在の表示内で下に移動
        空の項目を選択するため動かさない
        If (locSelected - 1) * MAXDISPLAY + itemSelected < MAXITEM Then
          itemSelected = itemSelected + 1
          selector.Top = itemBox(itemSelected).Top - GAP
          rItem(0).Top = rItem(whichrItem(itemSelected)).Top
          rItem(0).Left = locL - GAP
          rItem(0).Width = locW + 2 * GAP
          SetItemInfo
        End If
      ElseIf locSelected < MAXLOC Then
        リストの次の選択を表示
        itemSelected = 1
        locSelected = locSelected + 1
        RedoDisplay
      End If
    End Select
    rItem(0).Visible = True
  End Sub

Sub DoPreview ()

```

49

50

プレビューウィンドウおよびプレビューローケータを示す

Dim i As Integer 'counter

inPreview = True

他の要素は隠す

locator.Visible = False

selector.Visible = False

For i = 1 To MAXDISPLAY

itemBox(i).Visible = False

leftArrow(i).Visible = False

rightArrow(i).Visible = False

Next i

previewWin.Caption = "Getting preview..."

previewWin.ZOrder

previewWin.Visible = True

ShowPreview

End Sub

Sub DoSelect ()

現在選択された項目で作動

If Me Is TVlist Then

選択データをセットしTVへ行く

userStation = filterData("Station")

userStart = filterData("Start")

returnCode = TOTV

Me.Hide

Elseif Me Is MOVlist Then

'オーダゲームービー'メッセージを表示

sameFilter = True

TellUser "You would be asked to confirm your order of " &

filterData("Title")

Elseif Me Is SHOPlist Then

Select Case browsing

Case "stuff"

sameFilter = True

TellUser "You would be asked to confirm your order of " &

filterData("name")

Case "store"

filters(currDomain) = "(store code) = " & filterData("code")

Form\_Activate

Case "item"

filters(currDomain) = "(item code) = " & filterData("code")

Form\_Activate

Case "dept"

filters(currDomain) = "(dept code) = " & filterData("code")

Form\_Activate

End Select

End If

End Sub

51

52

```

Sub EndPreview ()
  通常のリストオペレーションに戻る
  Dim i As Integer 'counter

  previewWin.Visible = False
  inPreview = False
  locator.Visible = True
  selector.Visible = True
  previewWin.Top = displayList.Top
  RedoDisplay
End Sub

Sub Form_Activate ()
  Dim i As Integer 'counter
  Dim section As Integer   ロケータのロケーション数をカウント
  Dim NVisible As Integer  セクション内の可視形状を記録 (照合)
  Dim msg As String
  Static saveFilter As String
  Static saveView As String

  現在のフィルタに対して新しいフィルタをチェック
  If Not sameView Then sameView = (saveView = viewFilter)
  saveView = viewFilter
  If Not sameFilter Then sameFilter = (saveFilter = filters(currDomain))
  saveFilter = filters(currDomain)

  SetStatus currView(currDomain) & currFilter(currDomain), greyCOLOR

  If sameFilter And sameView Then
    すべてを最後と同じにキープ
    If newUser And Not Me Is SHOPlist Then
      popup.Caption = 'To change the category shown, press the 'Category'
button.'
      popup.Visible = True
      newUser = False
    End If
    RedoDisplay
  Else
    表示をクリーンアップ
    SetInfo "Selecting data, please wait...", GREY
    If MAXITEM = 0 Then
      previewWin.Caption = ""
      previewWin.Visible = False
    End If
    DoEvents
    If inPreview Then EndPreview

    For i = 1 To MAXDISPLAY
      itemBox(i).Caption = ""
    Next i

    For i = 1 To MAXITEM
      Unload ritem(i)
    Next i
  End Sub

```

53

54

```

新しいデータをフィルタリング
ApplyFilter
If MAXITEM = 0 Then
  'give "no matches" msg
  locator.Visible = False
  rItem(0).Visible = False
  For i = 1 To MAXDISPLAY
    MAXITEM = 0
    itemBox(i).Visible = False
    leftArrow(i).Visible = False
    rightArrow(i).Visible = False
  Next i
  previewWin.Caption = "No matches were found" & Chr(13)
  previewWin.Caption = previewWin.Caption & "Press 'Category' to change
the selection."
  previewWin.ZOrder
  previewWin.Visible = True
  itemSelected = 0
  locSelected = 0
Else
  リスト表示をやり直す
  filterData.MoveLast
  MAXITEM = filterData.RecordCount
  項目間の距離をセット
  rowOffset = (H - rItem(0).Height) / MAXITEM
  If rowOffset > rItem(0).Height + GAP Then rowOffset = rItem(0).Height +
GAP 'max distance
  rItem(0).Visible = False
  rItem(0).Top = T
  rItem(0).Left = locL + reducedEXTRA
  rItem(0).Width = locW - 2 * reducedEXTRA
  rItem(0).BackColor = itemCOLOR
  filterData.MoveFirst
  項目形状をサイズ合せし置く
  セクションブックマークをセット
  section = 0 'number of locator locations
  NVisible = MAXDISPLAY 'so first section will be marked correctly
  For i = 1 To MAXITEM
    Load rItem(i)
    If colorField <> "" Then
      rItem(i).BackColor = Color(Val(filterData(colorField)) Mod 9)
    End If
    NVisible = NVisible + 1
    rItem(i).Top = T + (i - 1) * rowOffset
    If NVisible > MAXDISPLAY Then
      新しいロケータロケーションを開始
      section = section + 1
      locStart(section) = i
      marker(section) = filterData.Bookmark
      NVisible = 1
    End If
    If Me Is TVlist Then
      縮小された項目の長さをセット

```



55

56

```

    If filterData("StartTS") < TS Then
        rItem(i).Left = rItem(1).Left - reducedEXTRA
        rItem(i).Width = rItem(i).Width + reducedEXTRA
    End If
    If filterData("FinishTS") > TS Then
        rItem(1).Width = rItem(i).Width + reducedEXTRA
    End If
    End If
    rItem(i).ZOrder
    rItem(i).Visible = True
    filterData.MoveNext
Next i
MAXLOC = section
locStart(section + 1) = MAXITEM + 1

minselectorの長さをセット (use rItem(0))
rItem(0).Left = locL - GAP
rItem(0).Width = locW + 2 * GAP

セレクトおよびロケータを初期化
itemSelected = 1
locSelected = 1
locator.Visible = True
rItem(0).BackColor = highlightCOLOR
itemBoxes内の記述をセット
RedoDisplay
End If
End If
End Sub

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)
    popup.Visible = False
    Select Case KeyCode
    Case Asc("Q")
        End
    Case B_BACK
        If Me Is SHOPlist And browsing = "item" Then
            我々が何を欲するかが確かでない
            returnCode = ALPHA
            Me.Hide
        Else
            returnCode = BACK
            Me.Hide
        End If
    Case B_HELP
        InvokeHelp
    Case B_PREVIEW
        If inPreview Then
            EndPreview
        Else
            DoPreview
        End If
    Case B_SELECT
        If MAXITEM > 0 Then DoSelect

```

```

57
Case B_UP
  If MAXITEM > 0 Then ChangeSel ("Up")
Case B_DOWN
  If MAXITEM > 0 Then ChangeSel ("Down")
Case B_RIGHT
  If Me Is TVlist Then
    returnCode = COMING
    Me.Hide
  End If
Case B_LEFT
Case B_PAGEUP
  If inPreview Then
    プレビューをスクロール
    If previewWin.Top < displayList.Top Then
      スクリーンの下方にプレビューウィンドウを移動
      previewWin.Top = previewWin.Top + displayList.Height
    End If
  Else
    If MAXITEM > 0 Then ChangeLoc ("Up")
  End If
Case B_PAGEDOWN
  If inPreview Then
    プレビューをスクロール
    If previewWin.Top + previewWin.Height > displayList.Top +
displayList.Height Then
      スクリーン上方にプレビューウィンドウを移動
      previewWin.Top = previewWin.Top - displayList.Height
    End If
  Else
    If MAXITEM > 0 Then ChangeLoc ("Down")
  End If
Case B_FILTER
  If Not Me Is SHOPlist Then
    returnCode = Filter
    Me.Hide
  End If
Case B_0
  returnCode = SHORTCUT
  Me.Hide
End Select
End Sub

Sub Form_Load ()
  Dim i As Integer 'counter
  Dim itemRoom

  カラーおよびフォントをセット
  itemBox(0).FontSize = largeFONT
  leftArrow(0).FontSize = largeFONT
  rightArrow(0).FontSize = largeFONT
  If displayMode = "PC" Then
    popup.FontSize = largeFONT
    previewWin.FontSize = largeFONT
  Else

```

59

60

```

    previewWin.FontSize = mediumFONT
    popup.FontSize = smallFONT
End If
rItem(0).BackColor = itemCOLOR
selector.FillColor = highlightCOLOR
displayList.FillColor = backgroundCOLOR
previewWin.BackColor = backgroundCOLOR
locator.FillColor = backgroundCOLOR
itemBox(0).BackColor = itemCOLOR
leftArrow(0).BackColor = itemCOLOR
rightArrow(0).BackColor = itemCOLOR
shpSlot.BorderColor = slotCOLOR
オブジェクトのサイズをスクリーンに合わせる
SizeAForm Me, DispTop, DispHeight, DispLeft, DispWidth
Me.Scale (0, 0)-(1000, 1000)
SizeAControl locator, T - GAP, H + GAP, locL - GAP, locW + 2 * GAP
SizeAControl shpSlot, T, H, locL + reducedEXTRA, locW - 2 * reducedEXTRA
SizeAControl displayList, T - GAP, H + GAP, displ, dispW
SizeAControl popup, dispW / 2, 4 * locW, dispW / 2, 4 * locW
CPlace 1, previewWin, displayList
locator.ZOrder
shpSlot.ZOrder
rItem(0).ZOrder
itemRoom = H / MAXDISPLAY
SizeAControl itemBox(0), T + (.5 * GAP), itemRoom - GAP, displ + EXTRA, dispW -
2 * EXTRA
SizeAControl patch(0), 50, (6.8 * itemBox(0).Height), (12.3 * itemBox(0).Width),
(7 * itemBox(0).Height)
If displayMode = "TV" Then
    patch(0).Left = 8.08 * itemBox(0).Width
    patch(0).Height = 3.7 * itemBox(0).Height
End If
SizeAControl leftArrow(0), T + (.5 * GAP), itemRoom - GAP, displ, EXTRA
SizeAControl rightArrow(0), T + (.5 * GAP), itemRoom - GAP, displ + dispW -
EXTRA, EXTRA
SizeAControl selector, T, itemRoom + GAP, displ, dispW
selector.ZOrder
For i = 1 To MAXDISPLAY
    'Load itemBox(i) 'Now created at design time--fixed number (6)
    itemBox(i).Visible = False
    CCopy itemBox(0), itemBox(i)
    patch(i).Visible = False
    CPlace 0, patch(i), patch(0)
    itemBox(i).Top = itemBox(0).Top + (i - 1) * itemRoom
    Load leftArrow(i)
    leftArrow(i).Top = itemBox(i).Top
    Load rightArrow(i)
    rightArrow(i).Top = itemBox(i).Top
Next i

リストデータをロードし表示をセットアップ
sameFilter = False
sameView = False
LoadData

```

61

```

Form_Activate
sameFilter = True
End Sub

```

```

Function ItemString () As String

```

```

    情報バーに用いられるようにmsgをセット

```

```

    Dim msg As String

```

```

    If Me Is TVlist Then

```

```

        msg = Format(filterData("Title")) & " on "

```

```

        msg = msg & StationString(filterData("Station")) & ", "

```

```

        msg = msg & TimeString(filterData("Start")) & " to "

```

```

        msg = msg & TimeString(filterData("Finish"))

```

```

    ElseIf Me Is MOVlist Then

```

```

        msg = Format(filterData("Title"))

```

```

        msg = msg & ", " & Format(filterData("Year"))

```

```

    ElseIf Me Is SHOPlist Then

```

```

        Select Case browsing

```

```

            Case "stuff"

```

```

                msg = Format(filterData("name")) & " - $" & Format(filterData("price"))

```

```

            End Select

```

```

    End If

```

```

    ItemString = msg

```

```

End Function

```

```

Sub LoadData: ()

```

```

    Dim refSnap As snapshot

```

```

    Dim refDate

```

```

    スナップショットとしてデータベースにロード

```

```

    If Me Is TVlist Then

```

```

        startTime = fakeToday + fakeTime

```

```

        Set DB = OpenDatabase(TVDB)

```

```

        Set refSnap = DB.CreateSnapshot("Reference")

```

```

        refSnap.FindFirst "Name = 'Date'"

```

```

        refDate = DateValue(refSnap("Date"))

```

```

        Set allData = DB.CreateSnapshot("Programs")

```

```

        時間のフィルタは実際には作動時に起こるだろう

```

```

        TS = (startTime - refDate) * 48

```

```

        allData.Filter = Overlap(TS, TS)

```

```

        Set allData = allData.CreateSnapshot()

```

```

    ElseIf Me Is MOVlist Then

```

```

        Set DB = OpenDatabase(MVDB)

```

```

        Set allData = DB.CreateSnapshot("Movies")

```

```

    ElseIf Me Is SHOPlist Then

```

```

        Set DB = OpenDatabase(SPDB)

```

```

        Set itemData = DB.CreateSnapshot("Items")

```

```

        Set storeData = DB.CreateSnapshot("Stores")

```

```

        Set deptData = DB.CreateSnapshot("Departments")

```

```

        Set stuffData = DB.CreateSnapshot("Stuff")

```

```

    End If

```

```

End Sub

```

```

Sub RedoDisplay ()

```

62

63

64

itemBoxes内の記述をロケータ内の項目に対応するようにセット  
 ' reposition locator and selector. update info box

```

Dim last As Integer
Dim i As Integer
Dim Index As Integer 'index of ritem

If MAXITEM = 0 Then Exit Sub
'第一の項目ロケーションを計算
filterData.Bookmark = marker(locSelected)

Index = locStart(locSelected)
For i = 1 To MAXDISPLAY
  If filterData.EOF Then
    'hide empty itemBox
    itemBox(i).Caption = ""
    itemBox(i).Visible = False
    leftArrow(i).Visible = False
    rightArrow(i).Visible = False
  Else
    whichrItem(i) = Index 'so we can highlight the correct ritem (reduced
item)
    If colorField <> "" Then itemBox(i).BackColor =
Color(filterData(colorField) Mod 9)
    itemBox(i).Caption = filterData(captionField)
    If Not inPreview Then itemBox(i).Visible = True
    If Me Is TVlist And Not inPreview Then
      'show arrows to reflect program length
      If filterData("StartTS") < TS Then
        leftArrow(i).BackColor = itemBox(i).BackColor
        leftArrow(i).Visible = True
      Else
        leftArrow(i).Visible = False
      End If
      If filterData("FinishTS") > TS Then
        rightArrow(i).BackColor = itemBox(i).BackColor
        rightArrow(i).Visible = True
      Else
        rightArrow(i).Visible = False
      End If
      'show color patch for subcategory
      patch(i).FillColor = Color(filterData("Category") Mod 9)
      patch(i).Visible = True
    End If
    last = i
    Index = Index + 1
    filterData.MoveNext
  End If
Next i

'blankが選択されないようにする
If itemSelected > last Then
  itemSelected = last
End If

```

65

66

```

表示の残りを固定
displayList.Height = H + 2 * GAP - (H / MAXDISPLAY * (MAXDISPLAY - last))
display list shrinks when fewer than MAXDISPLAY items displayed
selector.Top = itemBox(itemSelected).Top - GAP 'behind current itemBox
locator.Top = T + rowOffset * (locStart(locSelected) - 1)
locator.Height = last * rowOffset + rItem(0).Height - rowOffset 'height shrinks
when displayList shrinks
rItem(0).Top = rItem(whichrItem(itemSelected)).Top
SetItemInfo
End Sub

Sub SetItemInfo ()
情報ボックス内の現在の項目の情報を表示
Dim i As Integer '0 of records away from bookmark we need to go
Dim msg As String

選択されたレコードを見つける
filterData.Bookmark = marker(locSelected)
i = itemSelected
While (i > 1)
    filterData.MoveNext
    i = i - 1
Wend
情報を情報ボックスに入れる
SetInfo ItemString(), (itemBox(itemSelected).BackColor)
必要であればプレビューウィンドウを更新
If inPreview Then ShowPreview
End Sub

Sub ShowPreview ()
ビデオもしくはテキストプレビューを表示
' of the item selected
Dim msg As String
If Me Is MOVlist Then
    msg = filterData("Plot")
Elseif Me Is TVlist Then
    msg = filterData(captionField) & Chr(13)
    msg = msg & StationString(filterData("Station")) & Chr(13)
    msg = msg & CategoryString((filterData("Type")), (filterData("Category")))
Else
    msg = "This would be a video, still, or textual preview of "
    msg = msg & filterData(captionField)
    msg = msg & ""
End If
previewWin.Visible = False
previewWin.Caption = msg
CPlace 0, previewWin, displayList
previewWin.Visible = True
End Sub

Sub tmrBlink_Timer ()
BlinkControl.Visible = Not BlinkControl.Visible
End Sub

```

[0034]

67

68

```

'====メッセージフォームコード====
このフォームは情報を表示する為にヘルプおよびいくつかのリストにより用いられる
現在のフォームを一次的にカバーアップ
Option Explicit

Const GAP = 500

Sub Form_Activate ()
    textArea.Caption = userMsg
End Sub

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)
    Select Case KeyCode
    Case Else
        returnCode = KeyCode
        Me.Hide
    End Select
End Sub

Sub Form_Load ()
    カラー並びにフォントをセット
    Me.BackColor = itemCOLOR
    textArea.BackColor = itemCOLOR
    textArea.FontSize = largeFONT
    サイズをセット
    SizeAForm Me, DispTop, DispHeight, DispLeft, DispWidth
    SizeAControl textArea, GAP, DispHeight - 2 * GAP, GAP, DispWidth - 2 * GAP
    初期化
    textArea.Caption = ""
End Sub

```

20

[0035]

```

'====ローロデックスフォームコード====
このフォームはメインメニューおよびフィルタメニューを示す
不実行：フィルタボタンのカラーをタイプ/カテゴリーのカラーに対応させる
Option Explicit

Dim BlinkControl As Control  明滅する強調に対するポインタ
Dim parent As Integer       親カードの数
Dim current As Integer      現在のカードの数

特別なカード
注：カード内のフィルタカードの数その都度これは更新されるべきである
datafile changes
Const filterCARD = 1        TVフィルタメニュー
Const mfilterCARD = 68      ムービーフィルタメニュー
Const homeCARD = 96         メインメニュー
Dim lastCard As Integer     フィルタ内にある間の通常カードの数をホールド

Const MAXTITLE = 3  注意：MAXTITLEにおける変化はLoadGraphicsのコードにおける変化を必要とする
LoadGraphics
Const CARDSHIFT = 2.5  カード表示のためのもの---カード配置における変化量
Const MAXROWS = 3      カード表示のためのもの---ボタンの横の列の数
Const MAXCOLS = 3      カード表示のためのもの---カード上のボタンの縦の列の数

```

40

50



69

70

Const MAXCARD = 9 スクリーン上に表示されるカードの最大数  
 Const MAXITEM = 9 カード上のボタンの最大数

アクションコード：ボタン選択のために取るべきアクションを知らせる  
 actNEXTよりも大きいアクションには追加の入力が必要

Const actCOMING = 2  
 Const actNOW = 3  
 Const actNEXT = 5  
 Const actALPHASHOP = 6  
 Const actFILTER = 7  
 Const actALPHATV = 8  
 Const actALPHAMOV = 9  
 Const actDOMAIN = 10  
 Const actLATER = 11  
 Const actWEEK = 12  
 Const actWKEND = 13  
 Const actSCHED = 14  
 Const shortTVVIEW = 30  
 Const shortMVVIEW = 31  
 Const shortSPVIEW = 32  
 Const shortTVNOW = 35  
 Const shortTVFAV = 36  
 Const shortMVFAV = 37  
 Const shortSPFAV = 38  
 Const actMOVIE = 40  
 Const actSTORE = 50  
 Const actDEPT = 52  
 Const actMORE = 60  
 Const actNONE = 65  
 'for development only  
 Const actKEYS = 71  
 Const actTABS = 72

Sub Animate (direct As String, cardNo As Integer)

別のカードを開く、バックアップするもしくはボタンを選択するようにする

Dim index As Integer  
 Dim depth As Integer

DoEvents 'do not interrupt another animation  
 depth = Cards(current).level

Select Case direct

Case "Back"

If Cards(current).parent > 0 Then  
 CCopy sspCard(depth), sspCont

sspCont.Visible = True

Zoom 10, sspCont, sspItem(Cards(current).self)

DisplayCard (Cards(current).parent)

sspCont.Visible = False

End If

Case "Next"

index = Cards(current).selected

If index > 0 Then

CCopy sspItem(index), sspCont

sspCont.Visible = True

71

72

```

sspCont.BackColor = sspCard(0).BackColor
Zoom 10, sspCont, sspCard(depth)
DisplayCard cardNo
sspCont.Visible = False
End If
Case "Select"
index = Cards(current).selected
If index > 0 Then
CCopy sspItem(index), sspCont
sspCont.Visible = True
sspCont.BackColor = sspCard(0).BackColor
SizeAControl sspCard(0), 0, 500, 0, 500 'size of whole form
Zoom 10, sspCont, sspCard(0)
End If
End Select
End Sub

Sub BlinkStart (C As Control, vis)
  明滅するオブジェクトを可能とする
  Set BlinkControl = C
  BlinkControl.Visible = vis
  tmrBlink.Enabled = True
End Sub

Sub BlinkStop (vis)
  可視性をvisとしておき、明滅するオブジェクトをストップする
  tmrBlink.Enabled = False
  If BlinkControl Is Nothing Then
    何もしない
  Else
    BlinkControl.Visible = vis
  End If
  Set BlinkControl = Nothing
End Sub

Sub ButtonAction ()
  選択されたボタンに関連するアクションを行う
  Dim button As Integer
  Dim cardNo As Integer
  Dim msg As String

  button = Cards(current).selected 'item number of selected button on parent card
  cardNo = Cards(current).item(button) 'card number of selected button
  If button < 1 Then Exit Sub

  Select Case Cards(cardNo).actionCode
  Case actNONE
    非作動ボタン
    SetInfo "このオプションはまだ利用できない" greyCOLOR
  Case actNEXT
    次のカードを表示
    Animate "Next", Cards(current).item(button)
  Case actDOMAIN
    次のカードに行く前に現在のドメインを変更

```

73

74

```

currDomain = Val(Cards(cardNo).actionData)
setStatus Cards(cardNo).name, greyCOLOR
Animate "Next", Cards(current).item(button)
Case actMORE
  同じトッピング上の異なる選択を示す (currently same as actNEXT)
  Animate "Next", Cards(current).item(button)
Case actCOMING
  TVで行われるもののスケジュールを示す
  Animate "Select", 0
  sameFilter = False
  Set views(currDomain) = frmComing
  returnCode = SHOWVIEW
  Me.Hide
Case actNOW
  TVで今行われているものを示す
  currView(currDomain) = "TV 6:30pm : " 明らかにこれは現在の
時刻である
  Animate "Select", 0
  sameFilter = False
  sameView = True
  Set views(currDomain) = listFrm(currDomain)
  returnCode = SHOWVIEW
  Me.Hide
Case actLATER
  後日TVで行われるものを示す
  'currently non-functional
  'Animate "Select", 0
  'sameFilter = False
  'Set views(currDomain) = frmFriday
  returnCode = SHOWVIEW
  Me.Hide
Case actWEEK
  ウィークディのTVスケジュールを示す
  Animate "Select", 0
  sameFilter = False
  Set views(currDomain) = frmWkday
  returnCode = SHOWVIEW
  Me.Hide
Case actWKEND
  'ウィークエンドのTVスケジュールを示す
  'currently non-functional
  'Animate "Select", 0
  'sameFilter = False
  'Set views(currDomain) = frmWkend
  returnCode = SHOWVIEW
  Me.Hide
Case actSCHED
  TVスケジュールを示す
  'currently non-functional
  'Animate "Select", 0
  'sameFilter = False
  'Set views(currDomain) = frmSched
  returnCode = SHOWVIEW
  Me.Hide

```

75

76

```

Case actALPHASHOP
  ユーザからのストリング (文字列) を得る、ユーザのストリングで始まる項目を検索
  注: これはおそらく非常に難しいだろう
  Animate "Select", 0
  SetStatus "Shopping, " & Cards(cardNo).name, greyCOLOR
  msg = Cards(cardNo).actionData
  SetInfo msg, YELLOW
  Wait frmAlpha
  If returnCode <> BACK And userString <> "" Then
    sameFilter = False
    filters(currDomain) = "item"
    Set views(currDomain) = listFrm(currDomain)
    returnCode = SHOWVIEW
    Me.Hide
  End If
Case actALPHATV
  ユーザが番組タイトルを選択できるようにする
  Animate "Select", 0
  SetStatus "TV, " & Cards(cardNo).name, greyCOLOR
  returnCode = PICK
  Me.Hide
Case actALPHAMOV
  これは機能するようにフックアップされていないがおそらくこれは
  かなりactALPHATVであろう
  Animate "Select", 0
Case actFILTER
  TVビューへ新しいフィルタを送る
  filters(currDomain) = Cards(cardNo).actionData
  currFilter(currDomain) = Cards(cardNo).infotext
  sameFilter = False
  sameView = True
  returnCode = SHOWVIEW
  Me.Hide
Case actMOVIE
  ムービーリストを示す
  Animate "Select", 0
  If current > homeCARD Then
    ビュー (フィルタ) は変化している
    currView(currDomain) = Cards(cardNo).infotext
    viewFilter = Cards(cardNo).actionData
    sameView = False
    sameFilter = False
  Else
    カテゴリーは変化している
    currFilter(currDomain) = ": " & Cards(cardNo).infotext
    filters(currDomain) = Cards(cardNo).actionData
    sameView = True
    sameFilter = False
  End If
  Set views(currDomain) = listFrm(currDomain)
  returnCode = SHOWVIEW
  Me.Hide
Case actSTORE
  記憶のリストを示す

```

77

78

```

Animate "Select", 0
SetInfo "Choose a store:", TURQUOISE
SetStatus "Shopping", greyCOLOR
sameFilter = False
filters(currDomain) = "store"
Set views(currDomain) = listFrm(currDomain)
returnCode = SHOWVIEW
Me.Hide
Case actDEPT
    デパートメントからの商品を出す
    Animate "Select", 0
    SetStatus "Shopping, " & Cards(cardNo).infotext, greyCOLOR
    sameFilter = False
    filters(currDomain) = "dept"
    userString = Cards(cardNo).name
    Set views(currDomain) = listFrm(currDomain)
    returnCode = SHOWVIEW
    Me.Hide
Case shortTVVIEW
    最新のTVスケジュールまたはリストを示す
    Animate "Select", 0
    currDomain = TV
    If views(currDomain) Is Nothing Then
        Set views(currDomain) = frmComing
    End If
    sameFilter = True
    returnCode = SHOWVIEW
    Me.Hide
Case shortMOVVIEW
    最新のムービーのリストを示す
    Animate "Select", 0
    currDomain = MOVIE
    If views(currDomain) Is Nothing Then
        Set views(currDomain) = listFrm(currDomain)
    End If
    sameFilter = True
    returnCode = SHOWVIEW
    Me.Hide
Case shortSPVIEW
    最新のショッピング画面を示す
    Animate "Select", 0
    currDomain = SHOP
    If views(currDomain) Is Nothing Then
        Set views(currDomain) = listFrm(currDomain)
    End If
    sameFilter = True
    returnCode = SHOWVIEW
    Me.Hide
Case shortTVNOW
    今やっている(オンナウの)TV番組すべてを示す
    currFilter(currDomain) = "All Categories"
    currView(currDomain) = "TV 6:30pm : " 'obviously, this would be the current
time
    Animate "Select", 0

```

79

80

```

currDomain = TV
filters(currDomain) = ""
sameFilter = False
sameView = True
Set views(currDomain) = listFrm(currDomain)
returnCode = SHOWVIEW
Me.Hide
Case actKEYS
    発展についてのみ、ステイしない
    SetKeys Cards(cardNo).actionData
    SetStatus Cards(cardNo).infotext, itemCOLOR
    current = homeCARD
    DisplayCard current
Case actTABS
    発展についてのみ
    ToggleTabs
Case Else
    MsgBox "Bad action code for card " & Cards(cardNo).name
    Stop
End
End Select
End Sub

Sub ChangeSel (direct As String)
    ボタン操作を行う
    Dim n As Integer
    Dim last As Integer, Sel As Integer
    n = Cards(current).NItems
    last = Cards(current).selected
    If last = 0 Then Exit Sub

    If direct = "Right" Then
        重ね合わせて右に移動
        If last = n Then
            Sel = 1
        Else
            Sel = last + 1
        End If
    ElseIf direct = "Left" Then
        重ね合わせて左に移動
        If last = 1 Then
            Sel = n
        Else
            Sel = last - 1
        End If
    ElseIf direct = "Up" Then
        重ね合わせなしで上に移動
        If last > MAXCOLS Then
            Sel = last - MAXCOLS
        Else
            Sel = last
        End If
    ElseIf direct = "Down" Then
        重ね合わせなしで下に移動

```

81

```

      If last <= n - MAXCOLS Then
        Sel = last + MAXCOLS
      Else
        Sel = last
      End If
    Else
      MsgBox "Bad Direction"
    End
  End If

  Cards(current).selected = Sel
  UpdateSel
End Sub

```

```

Sub DisplayCard (index)

```

スクリーン上にメニューを表示するのに気を付ける

```

  Dim depth As Integer 'number of visible cards
  Dim i As Integer 'counter
  Dim ancestor As Integer 'card numbers

```

```

  current = index
  parent = Cards(current).parent
  depth = Cards(current).level

```

現在のあと(前)のカードを隠す

```

  For i = MAXCARD To depth + 1 Step -1
    sspTitle(i).Visible = False
    sspCard(i).Visible = False
  Next i

```

前のタブネームが正しく且つ可視状態であったことを確認

```

  ancestor = current
  For i = depth - 1 To 1 Step -1
    ancestor = Cards(ancestor).parent
    sspTitle(i).Caption = Cards(ancestor).name
    sspCard(i).Visible = True
    sspTitle(i).Visible = True
  Next i

```

現在のカードを示す

```

  sspTitle(depth).Caption = Cards(current).name
  sspCard(depth).Visible = True
  sspTitle(depth).Visible = True

```

現在のカード上にボタンを表示

```

  DisplayItems
End Sub

```

```

Sub DisplayItems ()

```

カード上にボタンを表示

```

  Dim Area As SSPanel
  Dim i As Integer
  Dim Dx, Dy, x, Y, w, h
  Dim NItems As Integer

```

83

84

```

NItems = Cards(current).NItems

Set Area = sspCard(Cards(current).level) 'this is a pointer, not a copy
  ボタンのサイズを計算
Dx = Area.Width * .9 / MAXCOLS
Dy = Area.Height * .9 / MAXROWS
w = Dx * .9
If w > 30 Then w = 30
h = Dy * .9
If h > 20 Then h = 20

sspBlinkBG.Visible = False
sspBlinkBG.ZOrder 0  前へもってゆく
  各ボタンを配置し示す
For i = 1 To NItems
  sspItem(i).Width = w
  sspItem(i).Height = h
  sspItem(i).Caption = Cards(Cards(current).item(i)).name
  If Cards(Cards(current).item(i)).actionCode = actNONE Then
    非動作ボタンをグレイ (灰色) にかえる
    sspItem(i).BackColor = greyCOLOR
  Else
    sspItem(i).BackColor = itemCOLOR
  End If
  x = Area.Left + .05 * Area.Width + (((i - 1) Mod MAXCOLS) + .5) * Dx
  Y = Area.Top + .05 * Area.Height + (Int((i - 1) / MAXCOLS) + .5) * Dy
  CenterItem sspItem(i), x, Y
  sspItem(i).ZOrder 0
  sspItem(i).Visible = True
Next i
  明滅をボタンよりも大きくする
CPlace 2, sspBlinkBG, sspItem(1)

  使用しないボタンを隠す
For i = NItems + 1 To MAXITEM
  sspItem(i).Visible = False
Next i
UpdateSel
End Sub

Sub Form_Activate ()
  別のフォームからリターンコードをチェック
  sspCont.Visible = False
  Select Case returnCode
    Case BACK
      If current < homeCARD Then current = lastCard
      SetStatus '矢印を使用して選択をするまたはキーボードを使用する' greyCOLOR
      DisplayCard current
      UpdateSel
    Case SHORTCUT
      current = homeCARD
      SetStatus '矢印を使用して選択をするまたはキーボードを使用する' greyCOLOR
      DisplayCard current

```



85

86

```

UpdateSel
Case FILTER
SetStatus "矢印を使用して選択するまたはキーボードを使用する", greyCOLOR
If current < homeCARD Then
DisplayCard current
Else
lastCard = current
Select Case currDomain
Case TV
DisplayCard filterCARD
Case MOVIE
DisplayCard mfilterCARD
Case SHOP
DisplayCard current
End Select
End If
UpdateSel
Case COMING
TVリスト画面からスケジュール画面を得る
Cards(current).selected = 2
sameFilter = False
Set views(currDomain) = frmComing
returnCode = SHOWVIEW
Me.Hide
End Select
End Sub

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)
Dim index As Integer
Dim n As Integer

Select Case KeyCode
Case B_BACK
メニュー階層に遡む
Animate "Back", 0
Case B_HELP
InvokeHelp
Case B_PREVIEW
userStation = 1
userStart = fakeTime
returnCode = TOTV
Me.Hide
Case B_SELECT
ボタンアクションを行う
ButtonAction
Case B_RIGHT
ChangeSel ("Right")
Case B_LEFT
ChangeSel ("Left")
Case B_UP
ChangeSel ("Up")
Case B_DOWN
ChangeSel ("Down")
Case B_PAGEUP

```

87

88

```
Case B_PAGEDOWN
操作なしで直接ボタンを選択するために数字のキーボードを使用する
Case B_1
  If Cards(current).NItems > 0 Then
    Cards(current).selected = 1
    UpdateSel
    ButtonAction
  End If
Case B_2
  If Cards(current).NItems > 1 Then
    Cards(current).selected = 2
    UpdateSel
    ButtonAction
  End If
Case B_3
  If Cards(current).NItems > 2 Then
    Cards(current).selected = 3
    UpdateSel
    ButtonAction
  End If
Case B_4
  If Cards(current).NItems > 3 Then
    Cards(current).selected = 4
    UpdateSel
    ButtonAction
  End If
Case B_5
  If Cards(current).NItems > 4 Then
    Cards(current).selected = 5
    UpdateSel
    ButtonAction
  End If
Case B_6
  If Cards(current).NItems > 5 Then
    Cards(current).selected = 6
    UpdateSel
    ButtonAction
  End If
Case B_7
  If Cards(current).NItems > 6 Then
    Cards(current).selected = 7
    UpdateSel
    ButtonAction
  End If
Case B_8
  If Cards(current).NItems > 7 Then
    Cards(current).selected = 8
    UpdateSel
    ButtonAction
  End If
Case B_9
  If Cards(current).NItems > 8 Then
    Cards(current).selected = 9
    UpdateSel
```

```

69
    ButtonAction
End If
Case B_0
    current = homeCARD
    DisplayCard current
Case Asc("Q")
    End
End Select
End Sub

Sub Form_Load ()
    フォント並びにカラーをセット
    sspCard(0).BackColor = backgroundCOLOR
    sspTitle(0).BackColor = backgroundCOLOR
    sspItem(0).BackColor = itemCOLOR
    sspBlinkBG.BackColor = highlightCOLOR
    sspItem(0).FontSize = mediumFONT
    sspCard(0).FontSize = mediumFONT
    sspTitle(0).FontSize = mediumFONT
    Me.BackColor = formCOLOR
    表示エリアにフィットさせる
    SizeAForm Me, dispTop, dispHeight, dispLeft, dispWidth
    Me.Scale (0, 0)-(100, 100)
    グローバルリターンコードを省略へセット
    returnCode = BACK
    ローロテックスのメニュー階層内で読み込む
    PopulateCards
    グラフィックなオブジェクトをロード
    LoadGraphics
    スクリーン上に現在のカードをセット
    DisplayCard homeCARD
End Sub

Sub LoadGraphics ()
    Dim i As Integer 'counter
    Dim tabHeight

    ボタンをロード
    For i = 1 To MAXITEM
        Load sspItem(i)
    Next i

    プロットタイプのカードを形作る
    sspCard(0).Top = sspItem(0).Height
    sspCard(0).Height = 100 - CARDSHIFT - sspCard(0).Top
    sspCard(0).Left = 2 * CARDSHIFT
    sspCard(0).Width = 100 - 4 * CARDSHIFT
    'shape prototype tab
    sspTitle(0).AutoSize = False
    sspTitle(0).Width = sspCard(0).Width / MAXTITLE * CARDSHIFT
    カードおよびタブをロードし形作る
    For i = 1 To MAXCARD
        Load sspCard(i)
        sspCard(i).Height = sspCard(i - 1).Height - CARDSHIFT
    Next i

```

91

92

```

sspCard(i).Top = sspCard(i - 1).Top + CARDSHIFT
sspCard(i).ZOrder
Load sspTitle(i)
sspTitle(i).Top = sspCard(i).Top - sspTitle(0).Height + 2
Select Case (i Mod MAXTITLE)
  注: これらのケースは異なるMAXTITLEには柔軟でない
Case 1
  sspTitle(i).Left = sspCard(i).Left
Case 2
  sspTitle(i).Left = sspCard(i).Left + sspCard(i).Width / 2 -
sspTitle(i).Width / 2
Case 0
  sspTitle(i).Left = sspCard(i).Left + sspCard(i).Width - sspTitle(i).Width
End Select
sspTitle(i).ZOrder
Next i
End Sub

```

#### Sub PopulateCards ()

このサブルーチンは定義として既定されたCARDFILEファイルからロードデータ内で読み取る。カードはファイル内のライン(カード)の数まで1から符号が付けられるであろう。すべての特別なカードはホームカードの前に来なければならない。そして直前の定義として名付けられる各カードレコードはレベル(整数)、選択された項目(整数)、名称(ストリング(文字列))、情報ストリング(文字列)およびアクションコード(整数)を有するべきである。アクションコードがactNEXTよりも大きい場合は、追加入力があるカードについて読み取られる

```

Dim last As Integer, parent As Integer
Dim selected As Integer
Dim index As Integer, itemNo As Integer
Dim level, title, text, action
Open CARDFILE For Input As #1

  トップレベルの親のダミーを作る
  index = 0
  Cards(index).name = "root"
  Cards(index).level = 0
  Cards(index).NItems = 0
  While Not EOF(1)
    last = index
    index = index + 1
    Input #1, level, selected, title, text, action
    Cards(index).level = level
    Cards(index).selected = selected
    Cards(index).name = title
    Cards(index).infotext = text
    Cards(index).ActionCode = action
    If action > actNEXT Then
      Input #1, action
      Cards(index).actionData = action
    End If
  
```

```

93
Cards(index).NItems = 0  項目数を初期化
If Cards(index).level = Cards(last).level + 1 Then
    最後の子供
    parent = last
ElseIf Cards(index).level <= Cards(last).level Then
    親を見つけるためにバックアップ
    Do While (Cards(index).level < Cards(last).level)
        最後のsiblingを見つける
        last = Cards(last).parent
    Loop
    parent = Cards(last).parent
Else ラベルが省略された、テキストファイルは間違え
    MsgBox "Bad level in text file."
    Stop
End
End If
Cards(index).parent = parent
    項目の親リストに自身を加える
    itemNo = Cards(parent).NItems + 1
    Cards(parent).NItems = itemNo
    Cards(parent).item(itemNo) = index
    Cards(index).self = itemNo
Wend
Cards(0).NItems = 1
Close #1
End Sub

Sub tmrBlink_Timer ()
    BlinkControl.Visible = Not BlinkControl.Visible
End Sub

Sub ToggleTabs ()
    タブ位置のオフセットをトグル (切り替える)
    Dim i As Integer
    Static offset
    If offset = 3.5 Then
        offset = 2
    Else
        offset = 3.5
    End If
    For i = 1 To 9
        sspTitle(i).Top = sspCard(i).Top - sspTitle(0).Height + offset
    Next i
End Sub

Sub UpdateSel ()
    正しい位置に明滅強調を置く情報バーを更新
    Dim i As Integer
    Dim x, Y
    Dim S As SSPanel
    Dim text As String
    Dim color

    BlinkStop False  明滅を消す

```

95

96

```

i = Cards(current).selected
If i > 0 Then 何かが選択される
    Set S = sspItem(i) Sはボタンに対するポインタ
    ボタンの中央を見つける
    x = S.Left + S.Width / 2
    y = S.Top + S.Height / 2
    'put blinker behind button
    CenterItem sspBlinkBG, x, y
    'resume blinking
    BlinkStart sspBlinkBG, True
End If
text = Cards(Cards(current).item(Cards(current).selected)).infotext
color = sspItem(Cards(current).selected).BackColor
SetInfo text, color
End Sub

Sub Zoom (n As Integer, C As Control, Dest As Control)
    Destをコントロールするサイズを変えるCをコントロールするようにする
    Dim i As Integer, j As Integer
    Dim dl, dw, dt, dh
    dl = (Dest.Left - C.Left) / n
    dw = (Dest.Width - C.Width) / n
    dt = (Dest.Top - C.Top) / n
    dh = (Dest.Height - C.Height) / n
    C.ZOrder
    C.AutoSize = False
    For i = 1 To n
        C.Move C.Left + dl, C.Top + dt, C.Width + dw, C.Height + dh
        C.Refresh
    Next i
End Sub

```

[0036]

\*\*\*\*\* SELECTフォームコード \*\*\*\*\*  
 このフォームは有効な入力のみを許容するアルファベットの入力における別の試みである  
 2つのテーブルを有するTVスタイルのデータベースによる。参照テーブルは  
 最初に使われる  
 アルファベットの各文字または記号または数字から始まるすべての項目のカウントを含む  
 ユーザはまず可能な開始文字のリストを提示され(第1のオンスクリーンのリスト内の  
 項目はそこにいくつかの文字を有するであろう)、開始文字が選択されると、  
 タイトルのテーブルから適合するエントリからスナップショットが作られる。  
 ユーザが見るリストはそれぞれ次の文字の有効な選択を有し、もしくは  
 理解するための最良の方法はコードを読み取る前にアクション内の  
 フォームを見ることである。

コードはムービーのリストなどの他のデータと共闘するために容易に修正される。  
 今のところ選ばれた文字によって他のすべてのものから区別できる特別なタイトルの  
 場合はその全タイトルを有する。

注: itemBoxes内に用いられた統一のとれていないフォントはクーリエニューである

```

Option Explicit
Dim DB As database 'the full database

Dim list(1000) As String 'the list of selection strings

```

97

98

```

Dim leaf(1000) As Integer n个项目がリーフの場合正、それ以外誤
Dim listEnd As Integer リスト内の最終の要素の数
Dim currPrefix As String 今までに選ばれた文字
Dim initialList As Integer このリストが項目ごとに複数の文字を持つ場合正しい
Dim BlinkControl As Control 現在明滅するオブジェクトはない使用されていない
Dim itemSelected As Integer 1からMAXDISPLAYまで
Dim locSelected As Integer 1からMAXCOCまで
Dim rowOffset 2つの連続する縮小された項目のトップ間の差異

```

## データベース

```

Dim allData As snapshot
Dim filterData As snapshot
Dim marker(100) As String 各MAXDISPLAY項目のブックマーク
Dim locStart(100) As Integer ロケータの開始のrItemインデックス

```

## 'display parameters

```

Const MAXDISPLAY = 6 クローズマップされた項目の数
Dim MAXITEM As Integer 全リスト内の項目の数
Dim MAXLOC As Integer ロケータポジションの数
Dim whichrItem(MAXDISPLAY) As Integer どのrItemが現在の表示にあるか
Const GAP = 10 'space around lists
Const EXTRA = 70 'room for longer programs
Const reducedEXTRA = 20 縮小された表現内の長めの番組のためのルーム
Const T = 50
Const H = 1000 - 2 * T
Const locL = 30 縮小されたリスト
Const locW = 100
Const dispL = locW + 2 * locL 表示リスト
Const dispW = 1000 - dispL - locL

```

```

Sub BlinkStart (C As Control, vis)

```

```

    Set BlinkControl = C
    BlinkControl.Visible = vis
    tmrBlink.Enabled = True

```

```

End Sub

```

```

Sub BlinkStop (vis)

```

```

    tmrBlink.Enabled = False
    If BlinkControl Is Nothing Then 'do nothing
    Else

```

```

        BlinkControl.Visible = True

```

```

    End If

```

```

    Set BlinkControl = Nothing

```

```

End Sub

```

```

Sub ChangeLoc (direct As String)

```

```

    ロケータによりページをめくるもしくは戻す

```

```

    Select Case direct

```

```

        Case "Up"

```

```

            If locSelected > 1 Then

```

```

                locSelected = locSelected - 1

```

```

                RedoDisplay

```

```

            End If

```

```

        Case "Down"

```

```

99
100
    If locSelected < MAXLOC Then
        locSelected = locSelected + 1
        RedoDisplay
    End If
End Select
End Sub

Sub ChangeSel (direct As String)
    リスト操作を実行
    Select Case direct
    Case "Up"
        If itemSelected > 1 Then
            現在表示されている項目内で上に移動
            itemSelected = itemSelected - 1
            selector.Top = itemBox(itemSelected).Top - GAP
            rItem(0).Top = locator.Top + rowOffset * (itemSelected - 1)
            SetItemInfo
        ElseIf locSelected > 1 Then
            リストの前のセクションを表示
            itemSelected = MAXDISPLAY
            locSelected = locSelected - 1
            RedoDisplay
        End If
    Case "Down"
        If itemSelected < MAXDISPLAY Then
            現在表示されている項目内で下に移動
            空の項目を驚かしたり選択したりしない
            If (locSelected - 1) * MAXDISPLAY + itemSelected < MAXITEM Then
                itemSelected = itemSelected + 1
                selector.Top = itemBox(itemSelected).Top - GAP
                rItem(0).Top = locator.Top + rowOffset * (itemSelected - 1)
                SetItemInfo
            End If
        ElseIf locSelected < MAXLOC Then
            リストの次のセクションを表示
            itemSelected = 1
            locSelected = locSelected + 1
            RedoDisplay
        End If
    End Select
End Sub

Sub DoSelect ()
    リーフ値で終了もしくは接頭部 (プレフィックス) のユーザの選択に基づいた新しいリストを作る
    Dim index As Integer
    Dim count As Integer
    Dim i As Integer
    Dim nextChar As String
    Dim looking As Integer 'boolean
    Dim title As String

    index = locStart(locSelected) + itemSelected - 1  選択された項目のリスト内の
    インデックス
    If leaf(index) Then

```



101

102

```

行われた選択、次の画面を示す
title = removeAmpersand(list(index))
filterData.FindFirst "SelectTitle = '" & title & "'"
If filterData.NoMatch Then
    Do 'prompt for different title until found
        注:これは起こらない、これがデータベースにある場合リストに
        あるのみ
        title = InputBox(title & " not found. Enter new title:", title)
        filterData.FindFirst "SelectTitle = '" & title & "'"
    Loop Until Not filterData.NoMatch
End If
userString = filterData("FullTitle")
Set views(TV) = frmWeek
sameFilter = False
returnCode = SHOWVIEW
Me.Hide
Else
    何かが起こっていることをユーザに知らせる
    itemBox(itemSelected).BackColor = greyCOLOR
    SetInfo "Loading data, please wait...", greyCOLOR
    DoEvents
    i = Len(list(index))
    currPrefix = ""
    If initialList Then
        currPrefix = list(index)
    Else
        接頭部 (プレフィックス) からフォーマットする下線をのぞく
        If i > 2 Then currPrefix = Left(list(index), i - 2)
        currPrefix = currPrefix & Right(list(index), 1)
    End If
    SetStatus "TV Titles starting with '" & currPrefix, greyCOLOR
    'construct new list
    If initialList Then
        リスト項目は特別であり、プレフィックスでない
        If index = 1 Then
            選択された記号または数字
            initialList = False
            filterData.Filter = "SelectTitle < 'A'"
            currPrefix = ""
        Else
            選択された文字のリスト
            listEnd = 0
            For i = 1 To Len(currPrefix)
                'strip out the letters (ignore commas) to make a new list
                If Mid(currPrefix, i, 1) >= "A" Then
                    listEnd = listEnd + 1
                    list(listEnd) = "&" & Mid(currPrefix, i, 1)
                    leaf(i) = False
                End If
            Next i
        End If
    Else
        新しいプレフィックスと合うようにデータを再度フィルタリングする
        filterData.Filter = "SelectTitle like '" & currPrefix & "...."

```

103

104

End If

すでにソートされたものと仮定されたデータ

If Not initialList Then

依然データから新しいリストを作成する必要がある

Set filterData = filterData.CreateSnapshot()

filterData.MoveFirst

listEnd = 0

For i = Asc(" ") To Asc("Z") 'space, punctuation, and letters

注: ひとつひとつを全て試すのではなく、次の項目のcharに

進む方法で修正されるべき

count = 0: looking = True

While Not filterData.EOF And looking

nextChar = Mid(filterData("SelectTitle"), Len(currPrefix) + 1,

1)

If nextChar = Chr(i) Or nextChar = LCase(Chr(i)) Then

count = count + 1

filterData.MoveNext

Else

looking = False

End If

Wend

Select Case count

Case 0 リストには加えない

Case 1 リーフエントリーをつくる

filterData.MovePrevious

listEnd = listEnd + 1

list(listEnd) = fixAmpersand(filterData("SelectTitle"))

leaf(listEnd) = True

filterData.MoveNext

Case Else ノンリーフエントリーをつくる

filterData.MovePrevious

listEnd = listEnd + 1

list(listEnd) = currPrefix &amp; "&amp;" &amp; Chr(i) 新たなcharにアンダーラインを引く

注: アンダーラインは何が違うかを強調するためのひとつの

メカニズムである

leaf(listEnd) = False

filterData.MoveNext

End Select

Next i

If filterData.RecordCount &lt;= MAXDISPLAY Then

もしそれらの全てがディスプレイに納まる時は リーフのみがその中に含まれる

ようにリストを作り直す

listEnd = 0

filterData.MoveFirst

While Not filterData.EOF

listEnd = listEnd + 1

list(listEnd) = fixAmpersand(filterData("SelectTitle"))

leaf(listEnd) = True

filterData.MoveNext

Wend

End If

End If

105

106

```

'display the newly created list
itemBox(itemSelected).BackColor = itemCOLOR 新しく作成されたリストを表示
initialList = False
If listEnd > 1 Then
    NewList
Else
    もしリストにひとつしかないときは自動的に項目を選択する
    locSelected = 1
    itemSelected = 1
    DoSelect
End If
End If
End Sub

Sub Form_Activate ()
    常にイニシャル項目から始める
    LoadData
    NewList
End Sub

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)
    Select Case KeyCode
    Case Asc("Q")
        End
    Case B_BACK
        注: リストの中にある特定の選択値からレベルをひとつ戻す rel from a particular choice in
        能力が必要か?
        B_BACKによってリストをひとつ戻り、次にもう一度押すことによってメニューに戻ることができる
        returnCode = BACK
        Me.Hide
    Case B_HELP
        InvokeHelp
    Case B_PREVIEW
    Case B_SELECT
        DoSelect
    Case B_UP
        ChangeSel ("Up")
    Case B_DOWN
        ChangeSel ("Down")
    Case B_RIGHT
    Case B_LEFT
    Case B_PAGEUP
        ChangeLoc ("Up")
    Case B_PAGEDOWN
        ChangeLoc ("Down")
    Case B_FILTER
    Case B_0
        returnCode = SHORTCUT
        Me.Hide
    End Select
End Sub

Sub Form_Load ()
    Dim i As Integer カウンタ

```

107

108

```

Dim itemRoom

色とフォントをセット
itemBox(0).FontSize = largeFONT
rightArrow(0).FontSize = largeFONT
rItem(0).BackColor = itemCOLOR
selector.FillColor = highlightCOLOR
displayList.FillColor = backgroundCOLOR
locator.FillColor = backgroundCOLOR
itemBox(0).BackColor = itemCOLOR
rightArrow(0).BackColor = itemCOLOR
shpSlot.BorderColor = slotCOLOR
分類してオブジェクトをスクリーンに置く
SizeAForm Me, DispTop, DispHeight, DispLeft, DispWidth
Me.Scale (0, 0)-(1000, 1000)
SizeAControl locator, T - GAP, H + GAP, locL - GAP, locW + 2 * GAP
SizeAControl shpSlot, T, H, locL + reducedEXTRA, locW - 2 * reducedEXTRA
SizeAControl displayList, T - GAP, H + GAP, dispL, dispW
locator.ZOrder
shpSlot.ZOrder
rItem(0).ZOrder
itemRoom = H / MAXDISPLAY
SizeAControl itemBox(0), T + (.5 * GAP), itemRoom - GAP, dispL + EXTRA, dispW -
2 * EXTRA
SizeAControl leftArrow(0), T + (.5 * GAP), itemRoom - GAP, dispL, EXTRA
SizeAControl rightArrow(0), T + (.5 * GAP), itemRoom - GAP, dispL + dispW -
EXTRA, EXTRA
SizeAControl selector, T, itemRoom + GAP, dispL, dispW
selector.ZOrder
For i = 1 To MAXDISPLAY
    Load itemBox(i)
    itemBox(i).Visible = False
    itemBox(i).Top = itemBox(0).Top + (i - 1) * itemRoom
    Load rightArrow(i)
    rightArrow(i).Top = itemBox(i).Top
Next i
End Sub

Sub LoadData ()
Dim refSnap As snapshot
Const MAXTOGETHER = MAXDISPLAY ひとつの項目ボックスに許容される文字数
Dim together

イニシャルセレクションリストを埋める
listEnd = 0
Set DB = OpenDatabase(TVTitles)
Set allData = DB.CreateSnapshot("Titles")

イニシャルリストを作成する
Set refSnap = DB.CreateSnapshot("Reference")
refSnap.MoveFirst
together = MAXTOGETHER 新たな項目の必要性を示す
While Not refSnap.EOF
    Select Case refSnap("Number")

```

109

110

```

Case 0
  リストに加えない
Case 1
  リーフエントリーを作成する
  listEnd = listEnd + 1
  allData.FindFirst "SelectTitle like " & refSnap("Letter") & "*"
  list(listEnd) = allData("SelectTitle")
  leaf(listEnd) = True
  together = MAXTOGETHER
Case Else
  If refSnap("Letter") = "*" Then
    listEnd = listEnd + 1
    list(listEnd) = "Symbol or Number"
    together = MAXTOGETHER
    'note: DoSelect relies on this entry being the first list item
  Else
    If together >= MAXTOGETHER Then
      listEnd = listEnd + 1
      list(listEnd) = refSnap("Letter")
      together = 1
    Else
      list(listEnd) = list(listEnd) & ", " & refSnap("Letter")
      together = together + 1
    End If
  End If
  leaf(listEnd) = False
End Select
refSnap.MoveNext
Wend
Set filterData = allData
initialList = True
End Sub

Sub NewList ()
  新たなリストのために表示を作り直す
  注: ItemSelectedは1以外の何かに初期化されなければならないか?
  Dim i As Integer カウンタ
  Dim section As Integer ロケーターロケーションの数を数える
  Dim msg As String

  'clear captions
  For i = 1 To MAXDISPLAY
    itemBox(i).Caption = ""
  Next i

  For i = 1 To MAXITEM
    Unload rItem(i)
  Next i

  MAXITEM = listEnd リストの中の項目の数

  MAXITEMに関連する減らされた項目の形とサイズをロードする
  rowOffset = (H - rItem(0).Height) / MAXITEM
  If rowOffset > rItem(0).Height + GAP Then rowOffset = rItem(0).Height + GAP

```

111

112

```

rItem(0).Visible = False
rItem(0).Top = T
rItem(0).Left = locL + reducedEXTRA
rItem(0).Width = locW - 2 * reducedEXTRA
rItem(0).BackColor = itemCOLOR
filterData.MoveFirst
項目の形を分類してブレースクション
ブックマークをセット
section = 0      ロケータロケーションの数
For i = 1 To MAXITEM
    Load rItem(i)
    rItem(i).Top = T + (i - 1) * rowOffset
    If ((i - 1) Mod MAXDISPLAY) = 0 Then
        新しいロケータロケーションを開始
        section = section + 1
        locStart(section) = i
    End If
    If Not leaf(i) Then
        rItem(i).Width = rItem(i).Width + reducedEXTRA
    End If
    rItem(i).ZOrder
    rItem(i).Visible = True
Next i
MAXLOC = section
locStart(section + 1) = MAXITEM + 1

minselectorの長さをカット
ミニセレクトとしてrItem(0)を使用
rItem(0).Left = locL - GAP
rItem(0).Width = locW + 2 * GAP

セレクトとロケータを初期化
itemSelected = 1
locSelected = 1
rItem(0).BackColor = highlightCOLOR

項目ボックス内にキャプションをセット
RedoDisplay
End Sub

Sub RedoDisplay ()
    項目ボックス内にキャプションをセットしロケータ内の項目に対応
    ロケータとセレクトのポジションをとり直し情報ボックス内に項目情報をセット

    Dim last As Integer デ스플레이内の最後の項目の数
    Dim i As Integer カウンタ
    Dim index As Integer リスト内の項目のインデックス

    index = locStart(locSelected)
    For i = 1 To MAXDISPLAY
        If index > MAXITEM Then
            空きの項目ボックスを隠す
            itemBox(i).Caption = ""
            itemBox(i).Visible = False
        End If
        index = index + 1
    Next i

```

```

113
    rightArrow(i).Visible = False
Else
    itemBox(i).Caption = list(index)
    If Not leaf(index) Then
        右矢印を出して全てのcapsを入れる
        rightArrow(i).Visible = True
        itemBox(i).Caption = UCase(list(index))
    Else
        rightArrow(i).Visible = False
    End If
    itemBox(i).Visible = True
    last = i 最後の有効な選択を記憶
    index = index + 1
End If
Next i

    ブランクが選択されることを許容しない
    If itemSelected > last Then
        itemSelected = last
    End If

    表示の残りを直す
    displayList.Height = H + 2 * GAP - (H / MAXDISPLAY * (MAXDISPLAY - last))
    selector.Top = itemBox(itemSelected).Top - GAP
    locator.Top = T + rowOffset * (locStart(locSelected) - 1)
    locator.Height = last * rowOffset + rItem(0).Height - rowOffset
    rItem(0).Top = locator.Top + rowOffset * (itemSelected - 1)
    rItem(0).Visible = True
    SetItemInfo
End Sub

Function removeAmpersand (oldText As String) As String
    それぞれの2つのアンバーサンドからひとつを除く
    Dim text As String
    Dim newText As String
    Dim i As Integer

    text = oldText
    newText = ""
    While InStr(text, "&&")
        i = InStr(text, "&&")
        newText = newText & Left(text, i)
        text = Right(text, Len(text) - (i + 1))
    Wend
    removeAmpersand = newText & text
End Function

Sub SetItemInfo ()
    情報ボックスの中に現在の項目に関連する項目を入れる
    Dim msg As String
    Dim index As Integer
    Dim F As snapshot

    If Me.Visible Then

```

[0037]

115

116

'====TVフォームコード===='

本フォームは、もし現在TVがonでないときにTV番組を映すもしくは録画することを要求

Option Explicit

Const GAP = 700

Sub Form\_Activate ()

Dim msg As String

Dim DB As database

Dim Programs As table

Dim startTime

Dim refSnap As snapshot

Dim refDate

Dim startTS, finishTS, nowTS

Set DB = OpenDatabase(TVDB)

Set refSnap = DB.CreateSnapshot("Reference")

refSnap.FindFirst "Name = 'Date'"

refDate = DateValue(refSnap("Data"))

Set Programs = DB.OpenTable("Programs")

Programs.Index = "ID"

Programs.Seek "=", userStation, userStart

注: ユーザステーションが有効なことをチェックする必要

If Programs.NoMatch Then

現在ユーザステーションにあるものが何でも示すようにシミュレート

nowTS = DateDiff("n", refDate, fakeToday + fakeTime) \ 30

Set refSnap = Programs.CreateSnapshot()

refSnap.FindFirst "Station = " &amp; Str(userStation)

refSnap.FindNext "FinishTS &gt; " &amp; Str(nowTS)

msg = "You are watching "

msg = msg &amp; Chr(13) &amp; Format(refSnap("Title"))

msg = msg &amp; " on " &amp; StationString(refSnap("Station"))

msg = msg &amp; Chr(13) &amp; Format(refSnap("Start"), "h:mm AM/PM")

msg = msg &amp; " to " &amp; Format(refSnap("Finish"), "h:mm AM/PM")

Else

番組がオンならば決定そうでなければ記録

startTS = DateDiff("n", refDate, Programs("Start")) \ 30

finishTS = DateDiff("n", refDate, Programs("Finish")) \ 30

nowTS = DateDiff("n", refDate, fakeToday + fakeTime) \ 30

nowTSはリアルタイムで動くように計算される

If startTS &lt;= nowTS And finishTS &gt;= nowTS Then

msg = "You are watching "

Else

msg = "The VCR is set to record "

End If

msg = msg &amp; Chr(13) &amp; Format(Programs("Title"))

msg = msg &amp; " on " &amp; StationString(Programs("Station"))

msg = msg &amp; Chr(13) &amp; Format(Programs("Start"), "h:mm AM/PM")

msg = msg &amp; " to " &amp; Format(Programs("Finish"), "h:mm AM/PM")

End If

textArea.Caption = msg

End Sub



117

```

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)
    Select Case KeyCode
    Case B_BACK
        returnCode = LASTVIEW
        注: もしメニュー (ローデックス) から来たときは不適切
        Me.Hide
    Case B_0
        returnCode = SHORTCUT
        Me.Hide

    Case Asc("Q")
        End
    Case Else
        returnCode = BACK
        Me.Hide
    End Select
End Sub

```

```

Sub Form_Load ()
    textArea.Caption = ""
    textArea.FontSize = largeFONT
    SizeAForm Me, 0, ScrHeight, 0, ScrWidth
    SizeAControl textArea, GAP, ScrHeight - 2 * GAP, GAP, ScrWidth - 2 * GAP
End Sub

```

【0038】

===== TVガイドフォームコード =====

一般的注意

メインプロシージャはフレームを示すことによって回るボールを動かし、全てのフォームをロードし、そしてローデックスメニューを示す。コントロールはフォームからフォームへリターンコード変数 (グローバル申告のリターンコードリスト参照) の使用を経て移動させられる。例えばfraTxtは、リターンコードをSHOWVIEWにセットし、それ自身を隠す。これによってfraFrameが活動するようになる。fraFrameはリターンコードを見て、現在の領域の画像フォームを示す。フォーム間のコミュニケーションは、ひとつのフォームのプロシージャは外部からアクセスできないことから様々な変数によって為される。

Option Explicit

.....

グローバル宣言

.....

データベース定数

Global Const CARDFILE = "c:\pctv\db\cards2.txt"

Global Const MVDB = "c:\pctv\db\plots.mdb"

Global Const SPDB = "c:\pctv\db\shopping.mdb"

Global Const TVDB = "c:\pctv\db\big.mdb"

Global Const TVTitles = "c:\pctv\db\titles.mdb"

Const CATDB = "c:\pctv\db\cats.mdb"

Dim typeTable As table TVタイプIDs

Dim catTable As table TVカテゴリIDs

Dim statTable As table ステーションIDs

Global fakeToday 日付保持定数

Global fakeTime 時間保持定数

Global displayMode As String PCかTVに'表示'をセット (フォントとグラフィックのサイズに影響する)

Global newUser As Integer ブール 正しいとき、エキストラヘルプを与える

Global ScrWidth, ScrHeight

Global DispTop, DispHeight, DispLeft, DispWidth フレーム内部でフォームに与えられる

表示エリア

カラー

Global Const highlightCOLOR = &amp;H8080FF 赤

Global Const backgroundCOLOR = &amp;H80FFFF イエロー

118

119

120

```

Global Const itemCOLOR = &HFFFFFF    ライトブルー
Global Const formCOLOR = &HFF0000    ダークブルー
Global Const whiteCOLOR = &H80000005  ホワイト
Global Const greyCOLOR = &HC0C0C0    グレー
Global Const blackCOLOR = &H04        ブラック
Global Const slotCOLOR = &H80000005    ホワイト
Global Const borderCOLOR = &HFF4      レッド
Global Const divideCOLOR = &HFFFF4    ホワイト
Global Color(10) メインに埋められた配列

```

#### フォントサイズ

```

Global Const smallFONT = 13.8
Global Const mediumFONT = 18
Global Const largeFONT = 24

```

#### 領域定数

```

Global Const MOVIE = 0
Global Const TV = 1
Global Const SHOP = 2

```

#### リストフォームの配列

```

Global listFrm(3) As Form
Global TVlist As New frmList
Global MOVlist As New frmList
Global SHOPlist As New frmList

```

#### フォーム間のコミュニケーション

```

Global currDomain As Integer 現在の領域
Global filters(3) As String 現在の領域フィルタのクエリストリングの配列
Global currFilter(3) As String フィルタのテキスト名
Global currView(3) As String 現在の画像のテキスト名（主に画像タイトルを変える
リストに用いる）
Global views(3) As Form 現在の領域ビューの配列 [TV coming or TV schedule,
for instance)
Global viewFilter As String 適切な画像を得るために必要なデータベース
フィルタ
注：この場合はムービーにのみ用いられ、おそらく配列に拡大される
Global userString As String ユーザによって選択されたストリング
Global userMsg ユーザに表示するためのメッセージストリング
Global userStation ユーザによって選択された放送局（ステーション）
Global userStart ユーザによって選択された時間

```

```

Global sameFilter As Integer ブールデータをリフィルタする必要がある時は正しい
Global sameView As Integer ブールディスプレイをやり直す必要がある時は正しい

```

```

リターンコードはfrmFrame又はfrmDexの両活動のためにどちらのアクションをとるかを決定
Global returnCode As Integer
Global Const BACK = 0
Global Const TOTV = 1
Global Const LASTVIEW = 2
Global Const SHORTCUT = 3
Global Const DONE = 4

```

121

```
Global Const FILTER = 5
Global Const COMING = 6
Global Const SHOWVIEW = 9
Global Const ALPHA = 10
Global Const PICK = 11
Global Const STARTUP = 12
```

```
.....
ローロデックス用の
タイプカードを定義
.....
```

```
Global Const MAXITEM = 9   カード上のボタンの最大数
```

```
ひとつのインデックスカードをスクリーン上に見せるように表示
Type Card
```

```
self As Integer   親上のセルフの項目数
level As Integer  根から割れたカードの数
name As String    ボタン/カード上に現れるテキスト
infotext As String 情報バー用のテキスト
actionCode As Integer 選択されたときに取られるアクション用のコード
actionData As String アクションように必要なエキストラ情報
parent As Integer  親カードの数
NItems As Integer カード上に見えるボタンの数
Item(MAXITEM) As Integer カードボインタの配列 (カード上のそれぞれのボタンにひとつ)
selected As Integer 選択されたボタンの数
```

```
End Type
```

```
MAXCARDSインデックスカードまでの配列
Global Const MAXCARDS = 1000
Global Cards(MAXCARDS) As Card
```

```
.....
リモートボタン
.....
```

```
サブSetKeysに与えられた値
```

```
Global B_BACK
Global B_HELP
Global B_PREVIEW
Global B_UP
Global B_DOWN
Global B_LEFT
Global B_RIGHT
Global B_SELECT
Global B_PAGEUP
Global B_PAGEDOWN
Global B_1
Global B_2
Global B_3
Global B_4
Global B_5
Global B_6
Global B_7
Global B_8
Global B_9
```

122

123

124

Global B\_0  
Global B\_FILTER

.....

色

.....

Global Const RED = &HFF&  
Global Const ORANGE = &H80FF&  
Global Const YELLOW = &HFFFF&  
Global Const GREEN = &H80FF80  
Global Const TURQUOISE = &HFFFF00  
Global Const BLUE = &HFF0000  
Global Const VIOLET = &HFF00FF  
Global Const WHITE = &HFFFFFF  
Global Const BLACK = &H0&  
Global Const GREY = &HC0C0C0

.....

・ ビジュアルベーシックファイルからの定数 ・

.....

・・ CONSTANT.TXTから

キーコード

Global Const KEY\_LBUTTON = &H1  
Global Const KEY\_RBUTTON = &H2  
Global Const KEY\_CANCEL = &H3  
Global Const KEY\_MBUTTON = &H4  
Global Const KEY\_BACK = &H8  
Global Const KEY\_TAB = &H9  
Global Const KEY\_CLEAR = &HC  
Global Const KEY\_RETURN = &HD  
Global Const KEY\_SHIFT = &H10  
Global Const KEY\_CONTROL = &H11  
Global Const KEY\_MENU = &H12  
Global Const KEY\_PAUSE = &H13  
Global Const KEY\_CAPITAL = &H14  
Global Const KEY\_ESCAPE = &H1B  
Global Const KEY\_SPACE = &H20  
Global Const KEY\_PRIOR = &H21  
Global Const KEY\_NEXT = &H22  
Global Const KEY\_END = &H23  
Global Const KEY\_HOME = &H24  
Global Const KEY\_LEFT = &H25  
Global Const KEY\_UP = &H26  
Global Const KEY\_RIGHT = &H27  
Global Const KEY\_DOWN = &H28  
Global Const KEY\_SELECT = &H29  
Global Const KEY\_PRINT = &H2A  
Global Const KEY\_EXECUTE = &H2B  
Global Const KEY\_SNAPSHOT = &H2C  
Global Const KEY\_INSERT = &H2D  
Global Const KEY\_DELETE = &H2E  
Global Const KEY\_HELP = &H2F

LBUTTON, RBUTTONとは接触しない

125

KEY\_AからKEY\_Zまではアスキーでいう'A'から'Z'に等しい  
KEY\_0からKEY\_9まではアスキーでいう'0'から'9'に等しい

```
Global Const KEY_NUMPAD0 = &H60
Global Const KEY_NUMPAD1 = &H61
Global Const KEY_NUMPAD2 = &H62
Global Const KEY_NUMPAD3 = &H63
Global Const KEY_NUMPAD4 = &H64
Global Const KEY_NUMPAD5 = &H65
Global Const KEY_NUMPAD6 = &H66
Global Const KEY_NUMPAD7 = &H67
Global Const KEY_NUMPAD8 = &H68
Global Const KEY_NUMPAD9 = &H69
Global Const KEY_MULTIPLY = &H6A
Global Const KEY_ADD = &H6B
Global Const KEY_SEPARATOR = &H6C
Global Const KEY_SUBTRACT = &H6D
Global Const KEY_DECIMAL = &H6E
Global Const KEY_DIVIDE = &H6F
Global Const KEY_F1 = &H70
Global Const KEY_F2 = &H71
Global Const KEY_F3 = &H72
Global Const KEY_F4 = &H73
Global Const KEY_F5 = &H74
Global Const KEY_F6 = &H75
Global Const KEY_F7 = &H76
Global Const KEY_F8 = &H77
Global Const KEY_F9 = &H78
Global Const KEY_F10 = &H79
Global Const KEY_F11 = &H7A
Global Const KEY_F12 = &H7B
Global Const KEY_F13 = &H7C
Global Const KEY_F14 = &H7D
Global Const KEY_F15 = &H7E
Global Const KEY_F16 = &H7F
```

```
Global Const KEY_NUMLOCK = &H90
```

```
Function CategoryString (typeCode As Integer, catCode As Integer) As String
    TV番組のカテゴリ用ユーザに読めるストリングを作る
```

```
    Dim msg As String
```

```
    msg = "Category: "
```

```
    タイプコードを見る
```

```
    typeTable.Index = "ID"
```

```
    typeTable.Seek "=", typeCode
```

```
    If typeTable.NoMatch Then
```

```
        msg = msg & typeCode
```

```
    Else
```

```
        msg = msg & typeTable("Name")
```

```
    End If
```

```
    msg = msg & ", " 'all on one line, replaced: Chr(13) & "Subcategory: "
```

127

128

```

    カテゴリーコードを見る
    catTable.Index = "ID"
    catTable.Seek "=", catCode
    If catTable.NoMatch Then
        msg = msg & catCode
    Else
        msg = msg & catTable("Name")
    End If
    CategoryString = msg
End Function

Sub CCopy (Cfrom As Control, Cto As Control)
    CfromコントロールからCtoまでの属性をコピー
    Cto.Caption = Cfrom.Caption
    Cto.BackColor = Cfrom.BackColor
    Cto.Top = Cfrom.Top
    Cto.Height = Cfrom.Height
    Cto.Left = Cfrom.Left
    Cto.Width = Cfrom.Width
    Cto.FontSize = Cfrom.FontSize
End Sub

Sub CenterItem (Item As Control, x, y)
    ポイントの周りのコントロールを中心に置く
    Item.Left = x - Item.Width / 2
    Item.Top = y - Item.Height / 2
End Sub

Sub CPlace (extra, Cfrom As Control, Cto As Control)
    CfromをCtoとおなじところに、エキストラの違いをつけてブレース
    Cfrom.Top = Cto.Top - extra
    Cfrom.Left = Cto.Left - extra
    Cfrom.Height = Cto.Height + 2 * extra
    Cfrom.Width = Cto.Width + 2 * extra
End Sub

Function DayString (d, length As String) As String
    与えられた日付と特定された長さに基づいて適切な曜日に
    スtringを戻す
    Select Case Weekday(d)
    Case 1
        If length = "long" Then
            DayString = "Sunday"
        Else
            DayString = "Sun"
        End If
    Case 2
        If length = "long" Then
            DayString = "Monday"
        Else
            DayString = "Mon"
        End If
    Case 3
        If length = "long" Then

```

```

129
    DayString = "Tuesday"
Else
    DayString = "Tue"
End If
Case 4
    If length = "long" Then
        DayString = "Wednesday"
    Else
        DayString = "Wed"
    End If
Case 5
    If length = "long" Then
        DayString = "Thursday"
    Else
        DayString = "Thur"
    End If
Case 6
    If length = "long" Then
        DayString = "Friday"
    Else
        DayString = "Fri"
    End If
Case 7
    If length = "long" Then
        DayString = "Saturday"
    Else
        DayString = "Sat"
    End If
End Select
End Function

Function fixAmpersand (text As String)
    アンダーラインをフォーマットする代わりにあらゆる'&'に'&&'を挿入するとアンバーサンドがプリントする
    Dim i As Integer
    Dim oldText As String
    Dim newText As String

    newText = ""
    oldText = text
    While InStr(oldText, "&")
        i = InStr(oldText, "&")
        newText = Left(oldText, i - 1) & "&&"
        oldText = Right(oldText, Len(oldText) - i)
    Wend
    fixAmpersand = newText & oldText
End Function

Sub InvokeHelp ()
    現在のローケーションにパラメータを加えるか各々のフォームにローカルInvokeHelpを与える
    各映像に限定され各ボタンには限定されない
    TellUser "Press Help (?) again for general help, or press any button on the
remote for help with that button."
    Select Case returnCode
        Case B_HELP

```

130

131

132

```

    TellUser "General Help:"
Case B_PREVIEW
    TellUser "Use the Preview button to see a video preview of the highlighted
selection."
Case B_BACK
    TellUser "Use the Back button to back up to the previous screen."
Case KEY_ESCAPE
    TellUser "Use the Shortcut key to get to the shortcut buttons."
Case B_SELECT
    TellUser "Use the select button to choose a highlighted option."
Case Else
    TellUser "This help screen is not written yet."
End Select
End Sub

Sub Main ()
    Dim i As Integer
    Dim DB As database
    Set DB = OpenDatabase(CATDB)
    Set typeTable = DB.OpenTable("Type")
    Set catTable = DB.OpenTable("Category")
    Set DB = OpenDatabase(TVDB)
    Set statTable = DB.OpenTable("Stations")
    SetKeys displayMode
    異なるリストフォームをセット
    Set listFrm(TV) = TVlist
    Set listFrm(MOVIE) = MOVlist
    Set listFrm(SHOP) = SHOPlist
    カラー配列をセット
    Color(0) = &HBF00      ティールグリーン
    Color(1) = &HFFF80      ライトブルー
    Color(2) = &HFFC0FF      ライトピンク
    Color(3) = &HFF80FF      ダークピンク
    Color(4) = &H80C0FF      ミディアムオレンジ
    Color(5) = &HC0FFC0      ライトリストグリーン
    Color(6) = &HFF8080      ロイヤルブルー
    Color(8) = &HFFC0C0      ラベンダー
    Color(7) = &HC0C0C      オークル
    日付と時間をセット
    fakeToday = CVDDate("6/12/94")
    fakeTime = CVDDate("6:30 PM")

    newUser = True

    フォームをスタートアップ
    frmFrame.Show
    DoEvents
    全てのフォームをここにロード
    Load frmDex
    Load frmAlpha
    Load frmTV
    Load frmMsg
    ムービーフォーム
    currDomain = MOVIE

```



133

```

viewFilter = "Year >= 1993"
currView(MOVIE) = "Recent Movies"
currFilter(MOVIE) = ": All Categories"
setStatus "Movies", greyCOLOR
Load listFrm(MOVIE)
  ショッピングフォーム
currDomain = SHOP
filters(SHOP) = ""
setStatus "Shopping, compact disks", greyCOLOR
Load listFrm(SHOP)
  TVフォーム
currFilter(TV) = "Basketball"
currDomain = TV
filters(TV) = "Category = 39"
userString = "Nova"
frmWeekをロード
listFrm(TV)をロード
frmCoiningをロード
frmVhdayをロード
frmSelectをロード

  メインメニューを示す
setStatus "Use arrows and select or use keypad.", greyCOLOR
frmDex.Show
End Sub

Function Overlap (beginTS, endTS) As String
  beginTSとendTSを含むその間のレンジにあるTV番組を探すための
  クエリースtringを作る
  Overlap = "(StartTS <= " & Str(endTS) & " And FinishTS >= " & Str(beginTS) & ")"
End Function

Sub SetInfo (text As String, Color)
  情報ボックステキストとカラーを更新
  Dim s As SSPanel
  Set s = frmFrame!sspInfo 'works as long as form is loaded
  s.BackColor = Color
  s.Caption = text
End Sub

Sub SetKeys (mode As String)
  キーボード又は'リモート'のキーマッピングをセット
  B_1 = KEY_NUMPAD7
  B_2 = KEY_NUMPAD8
  B_3 = KEY_NUMPAD9
  B_4 = KEY_NUMPAD4
  B_5 = KEY_NUMPAD5
  B_6 = KEY_NUMPAD6
  B_7 = KEY_NUMPAD1
  B_8 = KEY_NUMPAD2
  B_9 = KEY_NUMPAD3
  If mode = "TV" Then
    1-9を除くすべてのボタンのキーボードを用いる
    B_BACK = KEY_SUBTRACT
  
```

134

135

```

B_HELP = 18 このキーの名前が何かわからない
B_PREVIEW = KEY_ADD
B_UP = Asc("8")
B_DOWN = Asc("2")
B_LEFT = Asc("4")
B_RIGHT = Asc("6")
B_SELECT = Asc("5")
B_PAGEUP = KEY_DIVIDE
B_PAGEDOWN = Asc("0")
B_0 = KEY_MULTIPLY
B_FILTER = KEY_RETURN
Else
  B_BACK = KEY_F1
  B_HELP = KEY_F3
  B_PREVIEW = KEY_F2
  B_UP = KEY_UP
  B_DOWN = KEY_DOWN
  B_LEFT = KEY_LEFT
  B_RIGHT = KEY_RIGHT
  B_SELECT = KEY_RETURN
  B_PAGEUP = KEY_PRIOR
  B_PAGEDOWN = KEY_NEXT
  B_0 = KEY_NUMPAD0
  B_FILTER = KEY_F4
End If
End Sub

```

```

Sub SetStatus (text As String, Color)
  新たなメッセージを持つステータスバーを更新
  Dim s As SSPanel
  Set s = frmFrame!sspStatus ' (works as long as form is loaded)
  s.BackColor = Color
  s.Caption = text
End Sub

```

```

Sub SizeAControl (Item As Control, t, H, l, w)
  ひとつのコントロールのサイズ属性をセット
  Item.Top = t
  Item.Left = l
  Item.Height = H
  Item.Width = w
End Sub

```

```

Sub SizeAForm (frm As Form, t, H, l, w)
  ひとつのフォームのサイズ属性をセット
  frm.Top = t
  frm.Left = l
  frm.Height = H
  frm.Width = w
End Sub

```

```

Function StationString (s) As String
  局 (ステーション) 番号を見てストリングとして局名を返す
  statTable.Index = "ID"

```

136

137

138

```

statTable.Seek "=", s
If statTable.NoMatch Then
    MsgBox "illegal station ID " & s
    Stop
End If
StationString = statTable("Name")
End Function

```

```

Sub TellUser (message As String)
    キーがプレスされるまでスクリーン上にメッセージを表示する
    おそらく用いられない
    userMsg = message
    Wait frmMsg
End Sub

```

```

Function TimeLabel (t) As String
    30分の時間毎に空白ストリングを戻す
    1..12時度を戻す、もしくは
    Dim s As String
    s = Format(t, "hh:mm AM/PM")
    If Mid(s, 4, 2) = "30" Then
        TimeLabel = ""
    Else
        s = Format(s, "h AM/PM")
        AM/PMをはがす
        TimeLabel = Left(s, Len(s) - 3)
    End If
End Function

```

```

Function TimeString (aDate) As String
    日付をAM/PMのない12時間形式又はゼロで始まる形にフォーマットする
    Dim theTime As String
    theTime = Format(aDate, "hh:mm AM/PM")
    theTime = Left(theTime, 5) 'take just 'hh:mm' part
    If Left(theTime, 1) = "0" Then
        theTime = Right(theTime, 4)
    End If
    TimeString = theTime
End Function

```

```

Sub Wait (F As Form)
    ひとつのフォームをもうひとつが自ら隠れるまで待つことを許す
    F.Show
    While (F.Visible)
        DoEvents
    Wend
End Sub

```

[0039]

139

140

=====ウィークフォームコード=====

Option Explicit

'スタックドチャンネル'ビューがTVサーチあるいは他の最小のサーチ  
(アプライフィルタに修正が必要)に用いられる

```
Dim allData(8) As snapshot  時間内の全てのデータ
Dim filterData(8) As snapshot 画像における各日のスナップショット
Dim NDays As Integer      表示内の日にちの数
Dim NSlots As Integer      表示内のタイムスロットの数
Dim NProgs As Integer      表示内の番組の数      in display
Dim colorField As String   曜日カラーを決定するデータベースフィールド
                              (フィールドはひとつの整数を含まねばならない)
Dim inPreview As Integer   ブール、もし正しければプレビューが映されるべき
Dim refDate As Integer     データタイムスロット用のリファレンス日付
Dim slotsPerDay As Integer 1日に許されたスロットの数
Dim currDay As Integer     当時の数
Dim dayWidth As Integer    日付ラベルの幅
Dim lblHeight As Integer   日付ラベルの高さ
Dim infoHeight As Integer  特殊化された情報パネルの高さ
Dim timeHeight As Integer   時間ラベルの高さ
Dim startTime As Integer    ビュー用の開始時間
Dim TSBegin As Long         当日の最初のタイムスロット
Dim TSEnd As Long          当日の最後のタイムスロット
Dim TScurrent As Long       現在の番組のタイムスロット
```

Sub ApplyFilter ()

userStringのタイトルによる特別な番組のためにフィルタリング

Dim i As Integer 'counter

各日用にスナップショットを作る

For i = 1 To NDays

allData(i).Filter = "Title = '" &amp; userString &amp; "'"

Set filterData(i) = allData(i).CreateSnapshot()

filterData(i).Sort = "StartTS"

Set filterData(i) = filterData(i).CreateSnapshot()

Next i

End Sub

Sub ChangeSel (d As String)

ビュー操作を行う

Dim current 'as database marker

Dim success As Integer 'boolean

Dim s As Integer 'station number

Dim best

Dim TS As Long 'time slot

Dim F As snapshot

Dim aDay As Integer

Dim marker 'as bookmark

Dim arrows As String

値をセーブし、値を初期化

current = filterData(currDay).Bookmark

Set F = filterData(currDay)

s = F("Station")

TS = TScurrent

aDay = currDay

success = False

Select Case d

Case "Right"

141

142

```

同じ日の遅い時間に移動
F.FindNext "StartTS > " & Str(TS)
success = Not F.NoMatch
If success Then
    情報矢印が必要かどうかチェック
    TS = F("StartTS")
    F.MoveNext
    If Not F.EOF Then
        If F("StartTS") = TS Then
            infoArrows "down"
        Else
            infoArrows "none"
        End If
    Else
        infoArrows "none"
    End If
    F.MovePrevious
End If
Case "Left"
    同じ日の早い時間に移動
    F.FindPrevious "StartTS < " & Str(TS)
    success = Not F.NoMatch
    If success Then
        TS = F("StartTS")
        コラムの最上段に行く
        F.FindFirst "StartTS = " & Str(TS)
        TS = F("StartTS")
        情報矢印が必要かどうかチェック
        F.MoveNext
        If Not F.EOF Then
            If F("StartTS") = TS Then
                infoArrows "down"
            Else
                infoArrows "none"
            End If
        Else
            infoArrows "none"
        End If
        F.MovePrevious
    End If
Case "Down"
    遅い日付に移動し、当初のタイムスロットに近いように維持するよう試みる
    If NProgs < 1 Then Exit Sub 'do nothing if all snapshots empty
    aDay = aDay + 1: TS = TS + 48
    While Not success And aDay <= NDays
        Set P = filterData(aDay)
        F.FindFirst "StartTS > " & Str(TS)
        If F.NoMatch Then
            右に番組がなければ、左に何か番組がないか探す
            If Not F.EOF Then F.MoveLast
            If Not F.EOF Then
                success = True
                TS = F("StartTS")
            End If
        End If
    End If

```

143

144

```

Else
    右に番組をセーブし、タイムスロットを数え、左をチェック
    marker = F.Bookmark
    best = F("StartTS") - TS
    F.FindLast "StartTS <= " & Str(TS)
    IF F.NoMatch Then
        左に番組がなければ、番組を右に移す
        F.Bookmark = marker
        TS = TS + best
    Else
        当初のタイムスロットからの距離をチェック
        If TS - F("StartTS") > best Then
            右の番組最も近い
            F.Bookmark = marker
            TS = TS + best
        Else
            左の番組最も近い
            TS = F("StartTS")
        End If
    End If
    どちらにしても番組を発見
    success = True
End If
aDay = aDay + 1: TS = TS + 48
Wend
aDay = aDay - 1: TS = TS - 48
If success Then
    コラムの最上段にあることを確認
    F.FindFirst "StartTS = " & Str(TS)
    If F.NoMatch Then Stop 'how did we get a TS with no program in it?
    TS = F("StartTS")
    情報矢印が必要かどうかチェック
    F.MoveNext
    If Not F.EOF Then
        If F("StartTS") = TS Then
            infoArrows "down"
        Else
            infoArrows "none"
        End If
    Else
        infoArrows "none"
    End If
    F.MovePrevious
End If
Case "Up"
    早い日付に移動し、当初のタイムスロットに近いように維持するよう試みる .me:slot
    If NProgs < 1 Then Exit Sub 'do nothing if all snapshots empty
    aDay = aDay - 1: TS = TS - 48
    While Not success And aDay > 0
        Set F = filterData(aDay)
        F.FindFirst "StartTS > " & Str(TS)
        If F.NoMatch Then
            右に番組がなければ、左を見る
            If Not F.EOF Then F.MoveLast

```

145

146

```

    If Not F.EOF Then
        success = True
        TS = F("StartTS")
    End If
Else
    左に番組をセーブして、タイムスロットを数え、左を見る
    marker = F.Bookmark
    best = F("StartTS") - TS
    F.FindLast "StartTS <= " & Str(TS)
    If F.NoMatch Then
        左に番組がなければ、右に番組を移す
        F.Bookmark = marker
        TS = TS + best
    Else
        距離をチェック
        If TS - F("StartTS") > best Then
            右の番組最も近い
            F.Bookmark = marker
            TS = TS + best
        Else
            左の番組最も近い
            TS = F("StartTS")
        End If
    End If
    いずれにしても番組を発見
    success = True
End If
aDay = aDay - 1: TS = TS - 48
Wend
aDay = aDay + 1: TS = TS + 48
If success Then
    コラムの最上段にあることを確認
    F.FindFirst "StartTS = " & Str(TS)
    If F.NoMatch Then Stop 番組がその中不在のどうやってTSを得たのか?
    TS = F("StartTS")
    情報矢印が必要かどうかチェック
    F.MoveNext
    If Not F.EOF Then
        If F("StartTS") = TS Then
            infoArrows = "down"
        Else
            infoArrows = "none"
        End If
    Else
        infoArrows = "none"
    End If
    F.MovePrevious
End If
Case "Next"
    次の番組を、同じ時間と日付で探す
    F.MoveNext
    If Not F.EOF Then
        サクセスはまだ同じタイムスロットにあることを意味
        success = F("StartTS") = TS
    End If

```

147

148

```

End If
矢印をセット
If success Then
    F.MoveNext
    arrows = "up"
    If Not F.EOF Then
        If F("StartTS") = TS Then arrows = "both"
    End If
    F.MovePrevious
    infoArrows arrows
End If
Case "Prior"
    当初の番組を、同じ時間、日付で探す
    F.MovePrevious
    If Not F.BOF Then
        サクセスは未だ同じタイムスロットにあることを意味
        success = F("StartTS") = TS
    End If
    矢印をセット
    If success Then
        F.MovePrevious
        arrows = "down"
        If Not F.BOF Then
            If F("StartTS") = TS Then arrows = "both"
        End If
        F.MoveNext
        infoArrows arrows
    End If
Case "none"
    現在の番組にとどまり矢印を更新(スタートアップで用いられた)
    If Not F.EOF Then
        F.MoveNext
        arrows = "none"
        If Not F.EOF Then
            If F("StartTS") = TS Then arrows = "down"
        End If
        F.MovePrevious
        infoArrows arrows
    End If
End Select

If success Then
    更新
    TScurrent = F("StartTS")
    currDay = aDay
    DisplayProg
Else
    データベースポジションをリストア
    filterData(currDay).Bookmark = current
End If
    当日の開始、終了タイムスロットをセット
    TSBegin = DateDiff("n", refDate, (startTime + currDay * 1)) \ 30
    TSEnd = TSBegin + slotsPerDay - 1
End Sub

```



149

150

```

Sub DisplayProg ()
    現在の番組情報に情報ボックスをセットしポジションを強調する
    Dim F As snapshot
    Dim msg As String

    Set F = filterData(currDay)
    msg = StationString(F("Station")) & " : " & Format(F("Start"), "h:mm AM/PM")
    msg = msg & " to " & Format(F("Finish"), "h:mm AM/PM, ")
    msg = msg & Format(F("Title"))
    msg = msg & Chr(13) & "(episode info here)" & Format(F("Episode"))
    注: 現在のデータベースはエピソード情報を含まない
    SetInfo msg, Color(F(colorField) Mod 9)

    shpProg(0).Visible = False
    selector.Visible = False
    Position shpProg(0), F("StartTS"), F("FinishTS")
    CPlace 0, selector, shpProg(0)
    shpProg(0).Visible = True
    selector.Visible = True
End Sub

Sub DoPreview ()
    適切なプレビューメッセージとディスプレイを構成
    Dim msg As String

    msg = filterData(currDay)("Title")
    msg = msg & Chr(13) & "on " & StationString(filterData(currDay)("Station")) &
    Chr(13)
    msg = msg & CategoryString((filterData(currDay)("Type")),
    (filterData(currDay)("Category")))
    msg = msg & Chr(13) & DayString(Weekday(filterData(currDay)("Start")), "long")
    msg = msg & ", " & Format(filterData(currDay)("Start"), "mmmm d, yy h:mm AM/PM")
    msg = msg & Chr(13) & "to " & Format(filterData(currDay)("Finish"), "h:mm
    AM/PM")

    popup.Caption = msg
    SizeAControl popup, (lblTime(1).top + 1.5 * lblTime(1).Height), 12,
    (lblDay(1).Width), 45
    popup.Visible = True
    inPreview = True
End Sub

Sub DoSelect ()
    セレクション用のデータをセットしてTVに行く
    userStation = filterData(currDay)("Station")
    userStart = filterData(currDay)("Start")
    returnCode = TOTV
    Me.Hide
End Sub

Sub DrawProg (duplicates As Integer, index As Integer)
    デイスプレイに番組の形を描き、同一のタイムスロットに重複が
    あるかどうかをマークする

```

151

shpProg(0)は望ましいロケーションにあるべき

```
Dim above, below, side, wide
Const GAP = .3
```

```
above = shpProg(0).top
below = shpProg(0).Height
side = shpProg(0).Left
wide = shpProg(0).Width
ForeColor = blackCOLOR 'line color, thin black outline
FillStyle = 0 'solid
drawwidth = 1
```

Select Case duplicates

Case 0

番組を与えられた色で描く

fillColor = Color(index)

Line (side, above)-(side + wide, above + below - .5 \* GAP), , B

Case 1

番組をグレーで描きマークする

fillColor = greyCOLOR 重複を示す (それは異なる

(色である)

Line (side, above)-(side + wide, above + below - .5 \* GAP), , B

アイコンを描く

above = above + GAP

side = side + GAP

wide = 2 \* GAP

drawwidth = 2

Line (side, above)-(side + wide, above + wide)

Line (side, above + wide)-(side + wide, above)

Line (side, above + .5 \* wide)-(side + wide, above + .5 \* wide)

Line (side + .5 \* wide, above)-(side + .5 \* wide, above + wide)

Case Else

重複マークを再度描く必要はない

End Select

End Sub

Sub Form\_Activate ()

Dim i As Integer カウンタ

Static saveFilter As String

If saveFilter = userString Then sameFilter = True

saveFilter = userString

SetStatus "This Week: " &amp; userString, greyCOLOR

同じフォームでなければ、消去して週間スケジュールを再度描く

If Not sameFilter Then

Me.Cls

SetInfo "Loading program information...", GREY

shpProg(0).Visible = False

selector.Visible = False

infoArrows "none"

DoEvents

153

154

```

        ApplyFilter
        MakeDisplay
        sameFilter = True
    End If
End Sub

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)
    Dim index As Integer
    Dim n As Integer
    Select Case KeyCode
    Case Asc("Q")
        End
    Case B_BACK
        returnCode = BACK
        Me.Hide
    Case B_HELP
        sameFilter = True
        InvokeHelp
    Case B_PREVIEW
        If inPreview Then
            inPreview = False
            popup.Visible = False
        Else
            inPreview = True
        End If
    Case B_RIGHT
        ChangeSel ("Right")
    Case B_LEFT
        ChangeSel ("Left")
    Case B_UP
        ChangeSel ("Up")
    Case B_DOWN
        ChangeSel ("Down")
    Case B_SELECT
        If Not filterData(currDay).EOF Then DoSelect
    Case B_PAGEDOWN
        ChangeSel ("Next")
    Case B_PAGEUP
        ChangeSel ("Prior")
    Case B_FILTER
        frmSelect1に戻り新たなタイトルを選ぶ
        returnCode = PICK
        Me.Hide
    Case B_0
        returnCode = SHORTCUT
        Me.Hide
    End Select
    If inPreview Then
        DoPreview
    Else
        popup.Visible = False
    End If
End Sub

```

155

156

```

Sub Form_Load ()
Dim i As Integer

    フォームの色とフォントをセット
    Me.BackColor = formCOLOR
    shpProg(0).BackColor = BorderColor
    lblDay(0).BackColor = backgroundCOLOR
    selector.BorderColor = BorderColor
    dayLine(0).BorderColor = divideCOLOR
    lblTime(0).ForeColor = slotCOLOR
    shpSlot(0).BorderColor = slotCOLOR
    infoPanel.FontSize = mediumFONT
    If displayMode = "TV" Then
        lblDay(0).FontSize = smallFONT
        lblTime(0).FontSize = smallFONT
        popup.FontSize = mediumFONT
    Else
        lblDay(0).FontSize = largeFONT
        lblTime(0).FontSize = largeFONT
        popup.FontSize = largeFONT
    End If
    スタンド・情報ボックスを隠す
    SizeAForm Me, 0, DispTop + DispHeight, DispLeft, dispWidth
    スケールとサイズオブジェクトをセット
    NDays = 7
    NSlots = 48
    dayWidth = 4
    lblHeight = 2
    infoHeight = 6
    If displayMode = "TV" Then
        upArrow.Left = 8950
        downArrow.Left = 8950
        timeHeight = 2
        downArrow.top = 650
        upArrow.top = 150
    Else
        timeHeight = 1.5
        downArrow.top = 1525
    End If
    Me.Scale (0, 0)-(NSlots + dayWidth, NDays * lblHeight + 2 * timeHeight +
infoHeight)
    selector.BorderWidth = 1
    延長した情報パネルを置く
    SizeAControl infoPanel, 0, infoHeight, 0, (Me.ScaleWidth)
    infoPanel.Caption = ""
    infoPanel.Visible = True
    日付ラベルをサイドに沿って置く
    SizeAControl lblDay(0), lblHeight + infoHeight, lblHeight, 0, dayWidth
    For i = 1 To NDays
        Load lblDay(i)
        lblDay(i).Caption = DayString(i, "short")
        lblDay(i).top = (i - 1) * lblHeight + infoHeight + 2 * timeHeight
        lblDay(i).Visible = True
    Next i

```

157

158

```

AM/PMラベルの上に渡す
SizeAControl lblDay(0), infoHeight, timeHeight, dayWidth, NSlots
lblDay(0).Caption = "AM"
lblDay(0).Visible = True
時間ラベルの上に渡す
SizeAControl lblTime(0), (lblDay(0).Height) + infoHeight, timeHeight, 0, 2
For i = 1 To NSlots \ 2
    Load lblTime(i)
    lblTime(i).Caption = TimeLabel(DateAdd("h", (i - 1), fakeTODAY))
    lblTime(i).Left = 2 * i + 2
    lblTime(i).Visible = True
Next i
NProgs = 0
sameFilter = False
InputData
Form_Activate
End Sub

Sub infoArrows (direct As String)
    同一時間にもっと番組があることを示す情報ボックスに矢印を見る
    又は隠す
    Select Case direct
        Case "up"
            downArrow.Visible = False
            upArrow.Visible = True
        Case "down"
            upArrow.Visible = False
            downArrow.Visible = True
        Case "both"
            upArrow.Visible = True
            downArrow.Visible = True
        Case "none"
            upArrow.Visible = False
            downArrow.Visible = False
    End Select
End Sub

Sub InputData ()
    form_loadの部分
    データベースを開きallDataスナップショットを作成する

    Dim DB As database
    Dim RefSnap As snapshot
    Dim i As Integer

    Set DB = OpenDatabase(TVDB)

    リファレンス日付と局数を得る
    Set RefSnap = DB.CreateSnapshot("Reference")
    RefSnap.FindFirst "Name = 'Date'"
    refDate = DateValue(RefSnap("Date"))
    RefSnap.FindFirst "Name = 'NStations'"

    Set allData(0) = DB.CreateSnapshot("Programs")

```

159

160

```

startTime = refDate
TSBegin = 0
TSEnd = TSBegin + 48 - 1
For i = 1 To 7
    allData(0).Filter = Overlap(TSBegin + 48 * (i - 1), TSEnd + 48 * (i - 1))
    Set allData(i) = allData(0).CreateSnapshot()
Next i
Set allData(0) = Nothing    もう全てのデータを一緒にには必要としない
End Sub

Sub MakeDisplay ()
    スクリーン上にスケジュールディスプレイを作成
    Dim i As Integer 'counter
    Dim d As Integer 'day
    Dim TSlast As Integer 'last time slot affected
    Dim F As snapshot 'convenience
    Dim offset As Integer 'used twice: dayline offset & number of programs sharing a
    time slot

    水平の日付線を描く
    drawwidth = 2
    ForeColor = lblDay(0).BackColor
    offset = infoHeight + 2 * timeHeight
    For i = 0 To NDays
        Line (0, offset + i * lblHeight) - (52, offset + i * lblHeight)
    Next i

    番組の形を描く
    offset = 0 特定のタイムスロットがどの程度フルになっているかトラックを続ける
    colorField = "Category" 注: これは代わりに 'Type' であるべきでは?
    For d = 1 To NDays
        currDay = d
        TSlast = -1
        Set F = filterData(d)
        If Not F.EOF Then
            F.MoveFirst
            Do While Not F.EOF
                If F("StartTS") = TSlast Then
                    offset = offset + 1
                    DrawProg offset, -1
                Else
                    offset = 0
                    Position shpProg(0), F("StartTS"), F("FinishTS")
                    DrawProg offset, F(colorField) Mod 9
                    TSlast = F("StartTS")
                End If
                F.MoveNext
            Loop
            F.MoveFirst
        End If
    Next d

    要素を初期化する

```

```

161
d = 1
currDay = 1
While d <= NDays
    If filterData(d).EOF Then
        d = d + 1
    Else
        NProgs = 1  ただそれが0以上であることを確認するため
        currDay = d
        d = NDays + 1
    End If
Wend
shpProg(0).ZOrder
selector.ZOrder
If Not filterData(currDay).EOF Then
    TSCurrent = filterData(currDay)("StartTS")
    DisplayProg
    ChangeSel "none"
End If
End Sub

Sub Position (shape As Control, start, finish)
    番組の形を位置付ける
    Dim leftTS
    Dim rightTS
    Const smallGAP = .1

    タイムスロットスケールに変換する
    leftTS = start - 48 * (currDay - 1)
    rightTS = finish - 48 * (currDay - 1)
    左と幅をセット
    shape.Left = dayWidth + leftTS
    shape.Width = rightTS - leftTS + 1 - smallGAP
    一日の始まりでカットオフする
    If shape.Left < dayWidth Then
        shape.Width = shape.Width - (dayWidth - shape.Left)
        shape.Left = dayWidth
    End If
    上と高さをセット
    shape.Height = 2 - 2 * smallGAP
    shape.top = lblDay(currDay).top + smallGAP
End Sub

Sub SetInfo (msg As String, Color)
    グローバルSetInfoを無視して自分独自の情報パネルに書き込む
    infoPanel.BackColor = Color
    infoPanel.Caption = msg
End Sub

```

[0040]

163

164

```

==ワークディフォームコード====
Option Explicit
特定の時間における平日5日間のスケジュール
タイムスロットにガイドされるナビゲーションを用いる

Dim allData(8) As snapshot 'all data within time period
Dim filterData(8) As snapshot 画像内の各日のスナップショット
Dim NDays As Integer          ディスプレイ内の日付の数
Dim NSlots As Integer         ディスプレイ内のタイムスロットの数
Dim NProgs As Integer         ディスプレイ内の番組の数
Dim NStation As Integer       ディスプレイ内の局の数
Dim MaxStation As Integer     データベース内の局の合計数
Dim colorField As String      項目カラーを決定するデータベースフィールド
                                (フィールドは1つの整数を含むべき)
Dim inPreview As Integer      ブール、もしプレビューが映れば正しい
Const sideGap = .05          番組の開始と終了においてスペース
Const topGap = 4             時間ラベルと最初の番組形の間にスペース
Dim refDate As Date           データタイムスロット用のリファレンス日付
Const lblHEIGHT = 40         日付と時間ラベルの高さ (500スケールで)
Const MINProgWidth = .2      スロットの分類として番組の形の最小幅
Dim slotsPerDay As Integer    一日に許されたスロットの数
Dim currDay As Integer        当日の数

Dim startTime As Date         ディスプレイの開始日付と時間
Dim TSBegin As Long           最初のタイムスロット
Dim TSEnd As Long            最後のタイムスロット
Dim TScurrent As Long         現在のタイムスロット
Dim rowOffset As Integer     スケジュール上の(上の)列の間の距離

Sub ApplyFilter ()
クエリーストリングフィルタ (TV) にフィットするTV番組だけのデータセットを作る
局の数と色を決定するデータベースフィールドをセットする
Dim i As Integer             カウンタ

If InStr(filters(TV), "Station") Then
    NStation = 10
    注: 好みのチャンネルを映すためのより良いメカニズムが必要
    colorField = "Type"
Else
    NStation = MaxStation
    colorField = "Category"
End If
For i = 1 To NDays
    allData(i).Filter = filters(TV)
    Set filterData(i) = allData(i).CreateSnapshot()
Next i
End Sub

Sub ChangeSel (d As String)
Dim current, firstMatch As Date データベースマーカーとして
Dim success As Integer          ブール
Dim s As Integer               ステーション
Dim TS As Long                 タイムスロット
Dim F As snapshot
Dim aDay As Integer
Dim best As Integer

current = filterData(currDay).Bookmark
Set F = filterData(currDay)
s = F("Station")

```



165

166

```

TS = TScurrent
aDay = currDay
success = False

If d = "Right" Then
    すぐ右の、同じタイムスロットをチェック
    F.MoveNext
    If Not F.EOF Then
        success = F("Station") = s And F("StartTS") = TS
    End If
    If Not success Then
        タイムスロットを右にチェック
        shpSlot(TS - TSBegin + 1 + (currDay - 1) * slotsPerDay).FillStyle =
1'transparent
        lblTime(TS - TSBegin + 1 + (currDay - 1) * slotsPerDay).BackStyle =
0'transparent
        While aDay <= NDays And Not success
            While TS < TSEnd And Not success
                TS = TS + 1
                現在の局から下の局をチェック
                F.FindFirst Overlap(TS, TS) & "And Station >=" & s
                If F.NoMatch Then
                    現在の局から最も上の局をとる
                    F.FindLast Overlap(TS, TS) & "And Station <=" & s
                    success = Not F.NoMatch
                Else
                    この組み合わせをセーブし上の局がより近いかどうかチェック
                    success = True
                    best = F("Station") - s
                    firstMatch = F.Bookmark
                    当初の組合せをチェック
                    F.FindPrevious Overlap(TS, TS)
                    If F.NoMatch Then
                        当初の組合せがなければ第一の組合せでいく
                        F.Bookmark = firstMatch
                    Else
                        If s - F("Station") > best Then
                            第一の組合せはより近い
                            F.Bookmark = firstMatch
                        End If
                    End If
                End If
            Wend
            If Not success Then
                TSBegin = TSBegin + 48
                TSEnd = TSBegin + slotsPerDay - 1
                TS = TSBegin - 1
                aDay = aDay + 1
                Set F = filterData(aDay)
            End If
        Wend
    End If
ElseIf d = "Left" Then
    すぐ左の、同じタイムスロットをチェック

```

167

168

```

F.MovePrevious
If Not F.BOF Then
    success = F("Station") = s And F("FinichTS") = TS
End If
If Not success Then
    "初のタイムスロットをチェック
    shpSlot(TS = TSBEGIN + 1 + (currDay - 1) * slotsPerDay).FillStyle =
1'transparent
    lblTime(TS = TSBEGIN + 1 + (currDay - 1) * slotsPerDay).BackStyle =
0'transparent
    While aDay >= 1 And Not success
        While TS > TSBEGIN And Not success
            'for each day to left
            '今日の当初のスロットを探す
            TS = TS - 1
            F.FindFirst Overlap(TS, TS) & " And Station >= " & Str(s)
            If F.NoMatch Then
                '現在の局より少ないものがなければ、第一のものを探す
                F.FindLast Overlap(TS, TS)
                success = Not F.NoMatch
            Else
                success = True
                'これをマークしチェック
                firstMatch = F.Bookmark
                best = F("Station") - s
                F.FindPrevious Overlap(TS, TS) 'will be less than current
            End If
        End While
        station
        If F.NoMatch Then
            'より低いものがなければ、第一の組合せを続ける
            F.Bookmark = firstMatch
        Else
            If s = F("Station") > best Then
                '第一の組合せがより近かった
                F.Bookmark = firstMatch
            End If
        End If
    End While
    'try previous day
    If Not success Then
        aDay = aDay - 1
        TSBEGIN = TSBEGIN - 48
        TSEnd = TSBEGIN + slotsPerDay - 1
        TS = TSEnd + 1
        Set P = filterData(aDay)
    End If
End If
ElseIf d = "Down" Then
    'タイムスロット内を下に移動する
    '注：現在のタイムスロットで_begin_する番組でだけストップする
    'オプションを持つ必要はあるか？
    '(with exception of first time slot in each day, of course)
    F.Bookmark = Current
    F.FindNext "(" & Overlap(TS, TS) & " And Station <> " & Str(s) & ")"
    success = Not F.NoMatch
ElseIf d = "Up" Then

```

169

170

```

タイムスロット内をドに移動
現在のタイムスロットで begin_する番組だけストップする
オプションを持つ必要はあるか?
(with exception of first time slot in each day, of course)
F.Bookmark = current
F.FindPrevious "(" & Overlap(TS, TS) & " And Station <> " & Str(e) & ")"
success = Not F.NoMatch
ElseIf d = "Top" Then
F.FindFirst Overlap(TS, TS)
success = Not F.NoMatch
ElseIf d = "Bottom" Then
F.FindLast Overlap(TS, TS)
success = Not F.NoMatch
End If

If success Then
更新
TScurrent = TS
currDay = aDay
DisplayProg
Else
データ内にポジションをリストアする
filterData(currDay).Bookmark = current
End If
他の要素をリストアする
TSBegin = DateDiff("n", refData, (startTime + currDay - 1)) \ 30
TSEnd = TSBegin + slotsPerDay - 1
shpSlot(TScurrent - TSBegin + 1 + (currDay - 1) * slotsPerDay).FillStyle =
0'solid
lblTime(TScurrent - TSBegin + 1 + (currDay - 1) * slotsPerDay).BackStyle =
1'opaque
End Sub

Sub DisplayProg ()
現在の番組のロケーションを検索する
情報ボックスに現在の番組用の情報を入れる
Dim F As snapshot
Dim msg As String

Set F = filterData(currDay)
番組を検索
shpProg(0).Visible = False
selector.Visible = False
Position shpProg(0), F("Start"), F("Finish"), F("Station")
CPlace 0, selector, shpProg(0)
shpProg(0).Visible = True
selector.Visible = True

メッセージをセット
msg = StationString(F("Station")) & " - " & F("Title") & " - "
msg = msg & Format(F("Start"), "h:mm AM/PM")
msg = msg & " to " & Format(F("Finish"), "h:mm AM/PM")
SetInfo msg, Color(F(colorField) Mod 9)
End Sub

```

171

172

```

Sub DoPreview ()
  適切なプレビューメッセージとディスプレイを構成
  Dim msg As String

  msg = "Station: " & StationString(filterData(currDay)("Station"))
  msg = msg & Chr(13) & "Title: " & filterData(currDay)("Title") & Chr(13)
  msg = msg & CategoryString(filterData(currDay)("Type"),
(filterData(currDay)("Category")))
  msg = msg & Chr(13) & "Time: " & Format(filterData(currDay)("Start"), "mm d,yy
h:mm AM/PM")
  msg = msg & Chr(13) & "to " & Format(filterData(currDay)("Finish"), "h:mm
AM/PM")

```

```

  プレビューメッセージとポップアップを映す
  popup.Caption = msg
  popup.Top = lblTime(1).Top + 2 * lblTime(1).Height
  popup.Left = 2
  popup.Width = slotsPerDay * NDays - 3
  popup.Visible = True
  inPreview = True
End Sub

```

```

Sub DoSelect ()
  選択のためのデータをセットしTVに行く
  userStation = filterData(currDay)("Station")
  userStart = filterData(currDay)("Start")
  returnCode = TOTV
  Me.Hide
End Sub

```

```

Sub Form_Activate ()
  Dim i As Integer 'counter
  Static saveFilter As String

  If saveFilter = filters(currDomain) Then sameFilter = True
  saveFilter = filters(currDomain)
  SetStatus "Evening TV: " & currFilter(TV), greyCOLOR
  注: イブニングTVラベルは変数であろう
  If inPreview Then
    popup.Visible = False
    inPreview = False
  End If
  If newUser Then
    popup.Caption = "Press 'category' to change the kind of programs displayed."
    popup.Visible = True
    注: ポップアップがボタン操作と同様にタイマーによって消えるようにするべき
    newUser = False
  End If

  If sameFilter Then
    暗くなったタイムスロットをリストア
    If TScurrent > 0 Then

```

173

174

```

shpSlot(TScurrent - TSBegin + 1 + (currDay - 1) * slotsPerDay).FillStyle
= 0'solid
lblTime(TScurrent - TSBegin + 1 + (currDay - 1) * slotsPerDay).BackStyle
= 1'solid
End If
Else
  古い番組の形をアンロードしディスプレイをやり直す
  SetInfo "Loading program information...", GREY
  shpProg(0).Visible = False
  lblDay(0).Visible = False
  lblTime(0).Visible = False
  shpSlot(0).Visible = False
  selector.Visible = False
  For i = 1 To NProgs
    Unload shpProg(i)
  Next i
  ApplyFilter
  MakeDisplay
  sameFilter = True
End If
End Sub

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)
  注: 立ち去る前に常にブラックスロットを消すようにすれば、その後の画像に悪影響を
  及ぼさない
  Dim Index As Integer
  Dim n As Integer
  Select Case KeyCode
    Case Asc("Q")
      End
    Case B_BACK
      shpSlot(TScurrent - TSBegin + 1 + (currDay - 1) * slotsPerDay).FillStyle =
1'transparent
      lblTime(TScurrent - TSBegin + 1 + (currDay - 1) * slotsPerDay).BackStyle =
0'transparent
      returnCode = BACK
      Me.Hide
    Case B_HELP
      sameFilter = True
      InvokeHelp
    Case B_PREVIEW
      If inPreview Then
        popup.Visible = False
        inPreview = False
      Else
        inPreview = True
      End If
    Case B_RIGHT
      If Not filterData(currDay).EOF Then ChangeSel ("Right")
    Case B_LEFT
      If Not filterData(currDay).EOF Then ChangeSel ("Left")
    Case B_UP
      If Not filterData(currDay).EOF Then ChangeSel ("Up")
    Case B_DOWN

```

175

176

```

    If Not filterData(currDay).EOF Then ChangeSel ("Down")
    Case B_SELECT
        shpSlot(TScurrent - TSBegin + 1 + (currDay - 1) * slotsPerDay).FillStyle =
1'transparent
        lblTime(TScurrent - TSBegin + 1 + (currDay - 1) * slotsPerDay).BackStyle =
0'transparent
        If Not filterData(currDay).EOF Then DoSelect
        Case B_PAGEDOWN
            If Not filterData(currDay).EOF Then ChangeSel ("Bottom")
        Case B_PAGEUP
            If Not filterData(currDay).EOF Then ChangeSel ("Top")
        Case B_FILTER
            shpSlot(TScurrent - TSBegin + 1 + (currDay - 1) * slotsPerDay).FillStyle =
1'transparent
            lblTime(TScurrent - TSBegin + 1 + (currDay - 1) * slotsPerDay).BackStyle =
0'transparent
            returnCode = Filter
            Me.Hide
        Case B_0
            shpSlot(TScurrent - TSBegin + 1 + (currDay - 1) * slotsPerDay).FillStyle =
1'transparent
            lblTime(TScurrent - TSBegin + 1 + (currDay - 1) * slotsPerDay).BackStyle =
0'transparent
            returnCode = SHORTCUT
            Me.Hide
    End Select
    If inPreview Then
        DoPreview
    Else
        popup.Visible = False
    End If
End Sub

Sub Form_Load ()
    Dim d As Integer, i As Integer, n As Integer 'counters
    Dim t 'as time
    フォームの色とフォントをセット
    Me.BackColor = formCOLOR
    shpProg(0).BackColor = BorderColor
    lblDay(0).BackColor = backgroundCOLOR
    selector.BorderColor = BorderColor
    dayLine(0).BorderColor = divideCOLOR
    lblTime(0).ForeColor = slotCOLOR
    shpSlot(0).BorderColor = slotCOLOR
    If displayMode = "TV" Then
        lblDay(0).FontSize = smallFONT
        lblTime(0).FontSize = smallFONT
        popup.FontSize = mediumFONT
    Else
        lblDay(0).FontSize = largeFONT
        lblTime(0).FontSize = largeFONT
        popup.FontSize = largeFONT
    End If
    オブジェクトのスケールとサイズをセット

```

```

SizeAForm Me, DispTop, DispHeight, DispLeft, DispWidth
Me.Scale (0, 0)-(500, 500)
SizeAControl lblDay(0), 0, lblHEIGHT, 0, 500
SizeAControl lblTime(0), lblHEIGHT, lblHEIGHT, 0, 50
SizeAControl shpSlot(0), 2 * lblHEIGHT + .5 * topGAP, 500 - 2 * lblHEIGHT, 0, 50
SizeAControl popup, 250, 200, 250, 200
selector.BorderWidth = 1
dayLine(0).Y1 = 0
dayLine(0).Y2 = 500
init:変数
sameFilter = False
sameView = False
inPreview = False
NProgs = 0
NDays = 5 平日5日間
slotsPerDay = 6 3時間、6 30分スロット
NSlots = NDays * slotsPerDay
startTime = fakeToday + CDate("7:00 PM") 時間は変数であり動的にセットされる
activate
startTime = DateAdd("d", 2 - Weekday(startTime), startTime) startTimeを月曜日
(=2) にセットする
タイムスロットスケールをセットレバーマネントオブジェクトに置く
Me.ScaleWidth = NSlots
lblTime(0).Width = 1
shpSlot(0).Width = 1
For i = 1 To NDays
    日付ラベルを起きキャプションを入れる
    Load lblDay(i)
    SizeAControl lblDay(i), 0, lblHEIGHT, slotsPerDay * (i - 1), slotsPerDay
    lblDay(i).Caption = DayString(i + 1, "short")
    lblDay(i).Visible = True
Next i
For d = 1 To NDays
    For i = 1 To slotsPerDay
        n = (d - 1) * slotsPerDay + i
        タイムスロットディバイダを置く
        Load shpSlot(n)
        shpSlot(n).Move n - 1
        shpSlot(n).ZOrder
        shpSlot(n).Visible = True
        時間ラベルを置く
        Load lblTime(n)
        lblTime(n).Move n - 1
        lblTime(n).ZOrder
        t = DateAdd("n", 30 * (i - 1), startTime)
        lblTime(n).Caption = TimeLabel(t)
        時間キャプションは、それらが変換されることがあるため(sameViewが間違ったとき)
        動的にセットされる
        shpSlot(i).FillStyle = 1'transparent
        lblTime(i).BackStyle = 0'transparent
        lblTime(n).Visible = True
    Next i
    日付セパレータを置く、しかしまだ映さない
    If d < NDays Then

```

```

179
    Load dayLine(d)
    dayLine(d).X1 = d * slotsPerDay
    dayLine(d).X2 = d * slotsPerDay
End If
Next d
InputData
Form_Activate
sameView = True
End Sub

Sub InputData ()
    form_loadの部分
    データベースを開きallDataスナップショットを作成する

    Dim DB As database
    Dim RefSnap As snapshot
    Dim i As Integer

    Set DB = OpenDatabase("TVDB")
    データがすでに局によってソートされていると仮定し、スタート

    リファレンス日付と局の値を得る
    Set RefSnap = DB.CreateSnapshot("Reference")
    RefSnap.FindFirst "Name = 'Date'"
    refDate = DateValue(RefSnap("Data"))
    RefSnap.FindFirst "Name = 'NStations'"
    MaxStation = Val(RefSnap("Data"))

    Set allData(0) = DB.CreateSnapshot("Programs")

    各平日の固定された時間の全ての番組のスナップショットを作成する
    時間は定数であって、それらは活動時に作成されるであろう
    TSBegin = DateDiff("n", refDate, startTime) \ 30
    TSEnd = TSBegin + slotsPerDay - 1
    For i = 1 To 5
        allData(i).Filter = Overlap(TSBegin + 48 * (i - 1), TSEnd + 48 * (i - 1))
48 時間 スロット/日付
        Set allData(i) = allData(0).CreateSnapshot()
    Next i
    Set allData(0) = Nothing 'won't be needing everything since time is fixed
End Sub

Sub MakeDisplay ()
    フィルタされたデータから番組のビジュアルスケジュールを作成
    Dim d As Integer 日付
    Dim c As Integer カウンタ
    Dim F As snapshot 便宜性
    Dim hasProgs As Integer 番組のはいつた第一日を記憶

    映している時間をセット
    If Not sameView Then
        ここで時間ラベルを変更
    End If

```



181

182

```

番組の形を置く
hasProgs = 0
c = 0 shpProgsのinitカウント
On Error GoTo ErrorHandler もし割り当てるshpProgsがなくなったとき
For d = 1 To NDays
    currDay = d
    Set F = filterData(d)
    データ内の各TV番組のシェイプコントロールを作成
    If Not F.EOF Then
        F.MoveFirst
        Do While Not F.EOF
            Load shpProg(c + 1)
            c = c + 1 割当てが成功した後にのみ増加
            shpProg(c).BackColor = Color(F(colorField) Mod 9)
            Position shpProg(c). F("Start"), F("Finish"), F("Station")
            shpProg(c).ZOrder
            shpProg(c).Visible = True
            F.MoveNext
        Loop
        F.MoveFirst
        If hasProgs = 0 Then hasProgs = d 'remember the first day with programs
    End If
Next d
MoveOn:
On Error GoTo 0 'quit trapping errors internally
上部に見える日付ラインをつくる
For d = 1 To NDays - 1
    dayLine(d).ZOrder
    dayLine(d).Visible = True
Next d

要素を初期化する
NProgs = c
currDay = hasProgs
shpProg(0).ZOrder
selector.ZOrder
If currDay > 0 Then
    当日のタイムスロット開始数と終了数をセット
    TSBegin = DateDiff("n", refDate, startTime) \ 30 + 48 * (currDay - 1)
    TSEnd = TSBegin + slotsPerDay - 1
    TScurrent = TSBegin
    Set F = filterData(currDay)
    Do While TScurrent <= TSEnd
        F.FindFirst Overlap(TScurrent, TScurrent)
        If Not F.NoMatch Then
            DisplayProg
            Exit Do
        End If
        TScurrent = TScurrent + 1
    Loop
Else
    TSBegin = DateDiff("n", refDate, startTime) \ 30
    TSEnd = TSBegin + slotsPerDay - 1

```

```

183
    TScurrent = TSBEGIN
    currDay = 1
End If
shpSlot(TScurrent - TSBEGIN + 1 + (currDay - 1) * slotsPerDay).FillStyle =
0'solid
lblTime(TScurrent - TSBEGIN + 1 + (currDay - 1) * slotsPerDay).BackStyle =
1'opaque
Exit Sub

ErrorHandler:
If Err = 342 Then
    'ran out of room to allocate program shapes, quit drawing
    Resume MoveOn
Else
    Dim msg
    msg = Error & Chr(13) & "Resume or Cancel?"
    msg = InputBox(msg, "Error Correction", "Resume")
    If msg = "" Then Stop
    Resume MoveOn
End If
End Sub

Sub Position (shape As Control, start, finish, station)
    '番組の形をディスプレイ用に位置づける
    Dim relativeL, relativeW, dayStart
    Dim edge
    '日付と時間をNSlotスケールに位置付けるために変換する
    dayStart = startTime + currDay - 1
    relativeL = (start - dayStart) * 48
    relativeW = (finish - dayStart) * 48 - relativeL
    '形を日付の境界線において切り抜く
    If relativeL < 0 Then
        relativeW = relativeW + relativeL
        relativeL = 0
    End If
    If relativeW + relativeL > slotsPerDay Then relativeW = slotsPerDay - relativeL
    '左と形の幅をセット
    edge = (currDay - 1) * slotsPerDay
    shape.Left = relativeL + edge + sideGap
    shape.Width = relativeW - 2 * sideGap
    '最小幅を強制して番組が見えるようにする
    If shape.Width < MINProgWidth Then shape.Width = MINProgWidth
    '上部を同じによってセット
    '注: このスキームは、局が1..nと名付けられていることによってのみ機能する
    rowOffset = ((500 - 2 * lblHEIGHT - shpProg(0).Height) / NStation)
    shape.Top = shpSlot(0).Top + topCAP + (station - 1) * rowOffset
End Sub

```

【0041】以上、本発明によるTV番組の多量のスケジュールから見る番組を検索かつ選択する方法および装置を開示した。そして、本発明は上記の好ましい実施の形態に関して特に例示し記述したが、当業者が形態、仕様、用途上の様々な変更を行いうることは自明である。例えば、縮小された表現の個々の項目や様々なグリッド表示における様々なエントリのカラーのコーディングを、視聴者が番組を素早く選択するのに役立たせてもよい。別の例として、本発明によるセット・トップ・ボックスを一体化することによりテレビを改造することも、本発明の技術の範囲内である。したがって、添付した請求の範囲は、本発明の主旨および範囲から逸脱しない限りにおいて上記のような形態、仕様、用途の変更お用すべてをカバーするものである。

#### 【図面の簡単な説明】

【図1】セット・トップ・ボックスを介して選択されるべき番組を搬送するケーブルに接続されたテレビと、その番組を選択するためのコントローラを図示した説明図

である。

【図2】セット・トップ・ボックスの簡略化したブロック図である。

【図3】図1に示されたコントローラを図示した説明図である。

【図4】最初の選択用の対話式表示を図示した説明図である。

【図5】第2段階の選択用の対話式表示を図示した説明図である。

【図6】第3段階の選択用の対話式表示を図示した説明図である。

【図7】第1段階の選択用の問い合わせ表示を図示した説明図である。

【図8】第2段階の選択用の問い合わせ表示を図示したものである。

【図9】第3段階の選択用の問い合わせ表示を図示した説明図である。

【図10】スポーツ関係、全ておよびオンナウ分類基準

に合った番組のサブグループを示すディスプレイを图示した説明図である。

【図11】強調された番組のプレビューを有するウインドウと共動してスポーツ関係、全オンナウ分類基準に合った番組のサブグループを示すディスプレイの説明図である。

【図12】図5と同様の、第2段階の選択用の対話式表示を示すディスプレイの説明図である。

【図13】非常に多くの番組データ項目を縮小した表現に示した2次元対話式グリッド表示を图示した説明図である。

【図14】図9と同様の、第3段階の選択用の問い合わせ表示を图示した説明図である。

【図15】処理しやすい数のデータ項目にフィルタリングされた2次元の対話式グリッド表示を图示した説明図である。

【図16】第1の英数字の対話式表示を图示したものである。

【図17】強調された対話エリアが異なった位置にある以外は図14と同じ表示の説明図である。

【図18】第2の英数字の対話式表示を图示した説明図である。

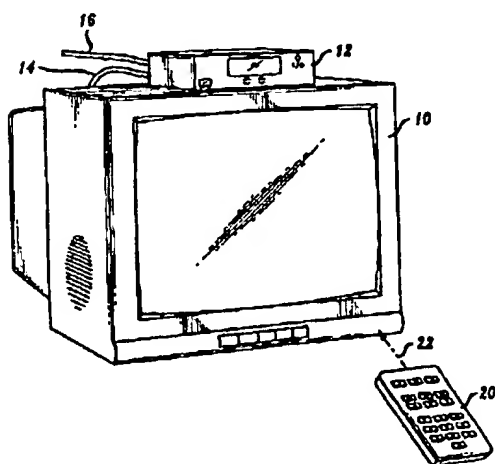
【図19】強調された対話エリアが異なった位置にある以外は図16と同じ表示の説明図である。

【図20】第3の英数字の対話式表示を图示したものである。

【図21】強調された対話エリアが異なった位置にある以外は図18と同じ表示の説明図である。

【図22】第3の英数字の対話式表示を图示した説明図である。

【図1】

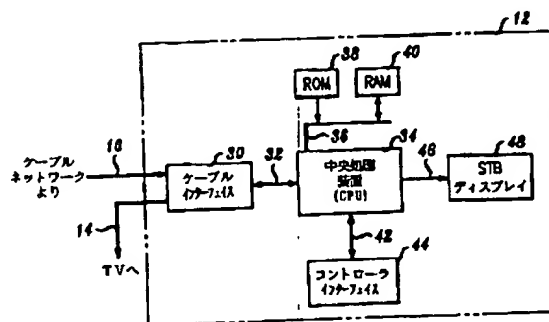


【図23】縦横の列の交点の3次元の論理スタックが多重エントリを有する状態の2次元対話式表示を图示した説明図である。

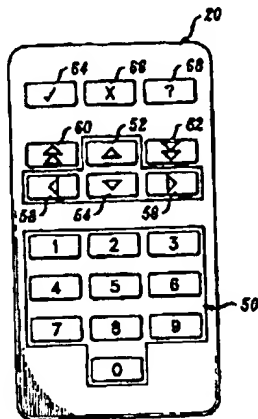
【符号の説明】

- 10 テレビ (TV)
- 12 セット・トップ・ボックス (STB)
- 14 相互接続ケーブル
- 16 ケーブル
- 20 コントローラ
- 22 光リンク
- 30 ケーブルインターフェイス
- 32 双方向バス
- 34 CPU
- 36 メモリバス
- 38 ROM
- 40 RAM
- 42 バス
- 44 コントローラインターフェイス
- 48 STBディスプレイ
- 50 キーパッド
- 52 上向き矢印
- 54 下向き矢印
- 56 右指示矢印
- 58 左指示矢印
- 60 2重上向き矢印
- 62 2重下向き矢印
- 64 セレクト (レ) ボタン
- 66 キャンセル (X) ボタン
- 68 問い合わせ (?) ボタン

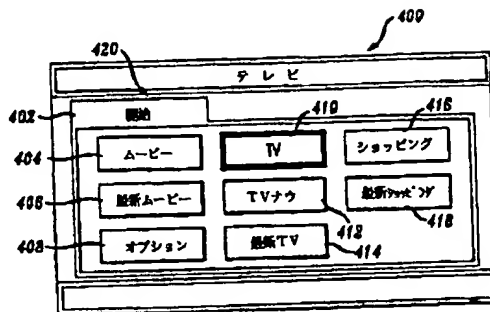
【図2】



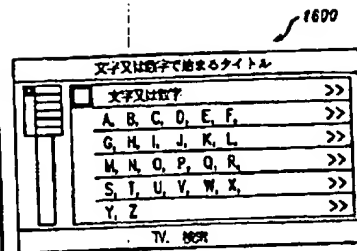
【図3】



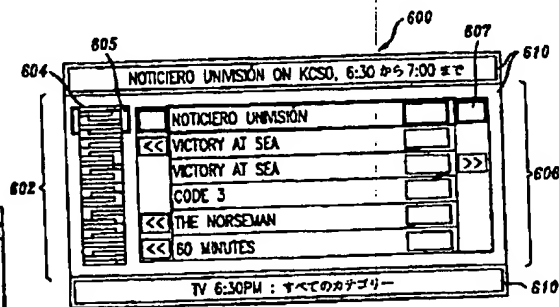
【図4】



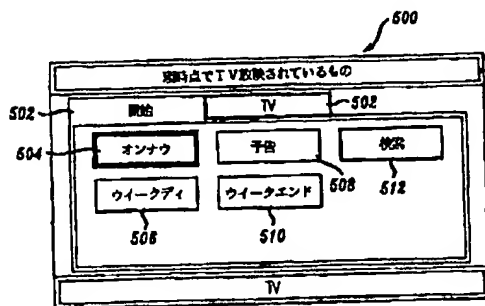
【図16】



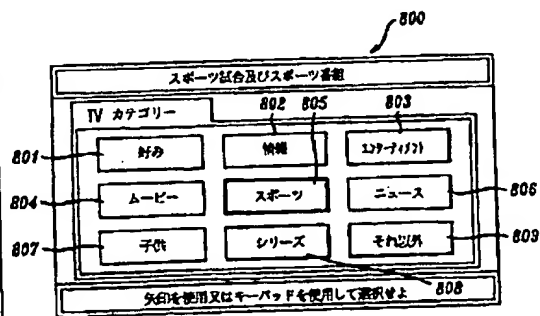
【図6】



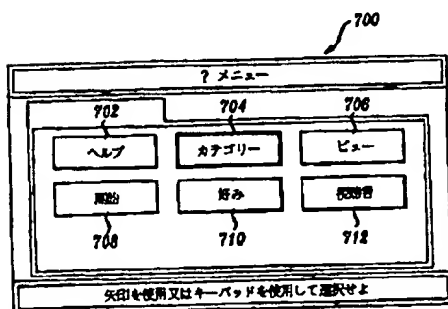
【図5】



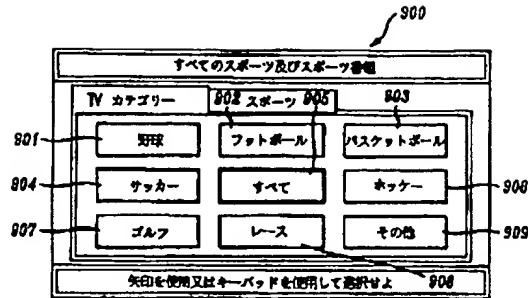
【図8】



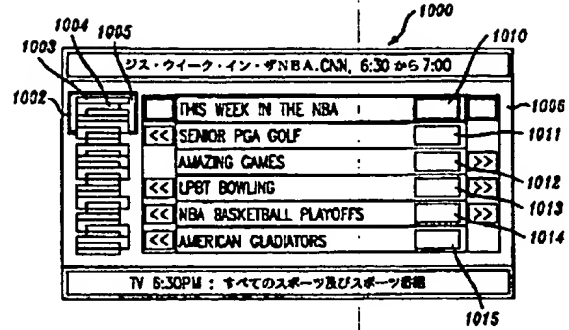
【図7】



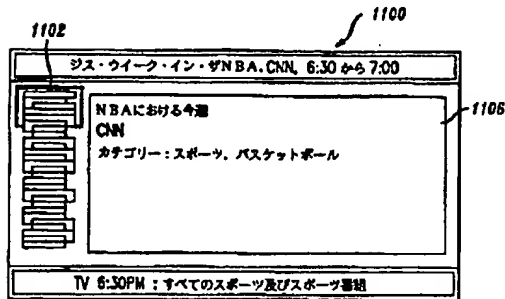
【図9】



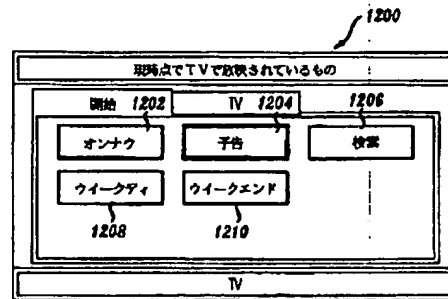
【図10】



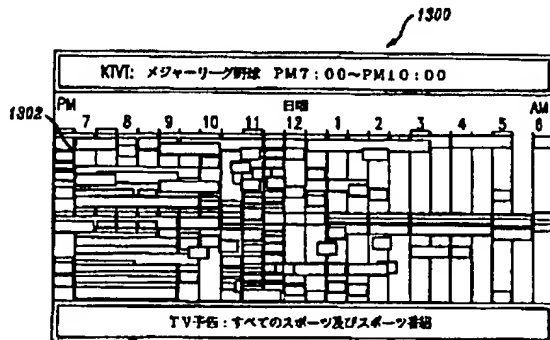
【図11】



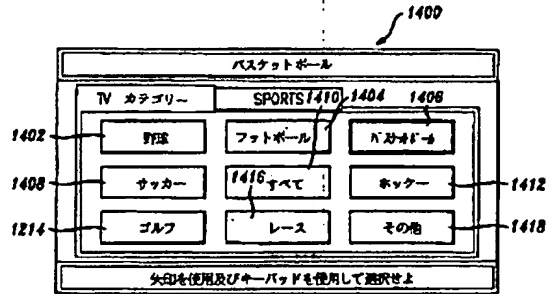
【図12】



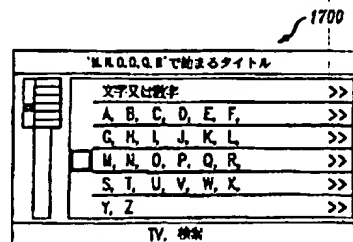
【図13】



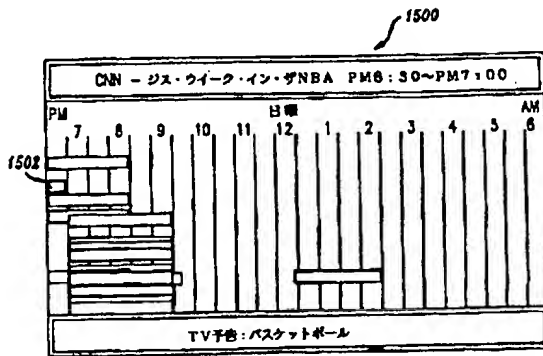
【図14】



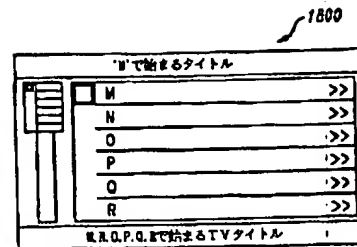
【図17】



【図15】



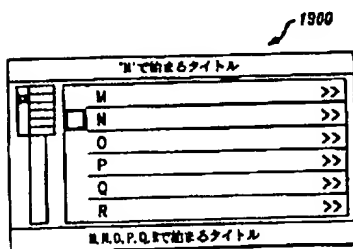
【図18】



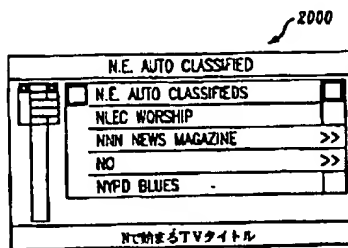
【図21】



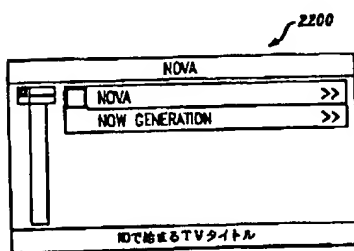
【図19】



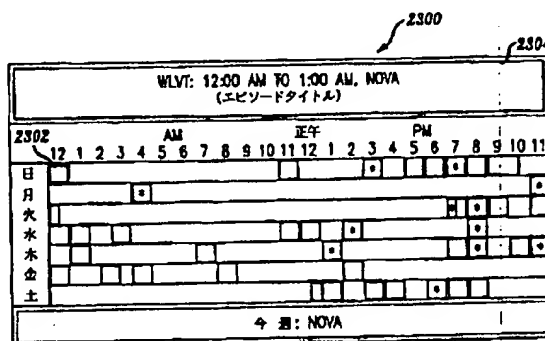
【図20】



【図22】



【図23】



フロントページの続き

(72)発明者 ピーター アンドリュー マタガ  
アメリカ合衆国 60563 イリノイズ, ネ  
イバーヴィル, イロクオイス アヴェニュー  
— 928

(72)発明者 レベッカ アン ウォルポール  
アメリカ合衆国 97330 オレゴン, コア  
ヴァリス, ノースウエスト ポーク アヴ  
エニュー 1820

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS

☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☒ FADED TEXT OR DRAWING

☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☒ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**